Revista de Usuarios Amstrad

Publicación electrónica del Amstrad CPC - Año II - Número 2 - Junio 2007

El juego que reta a tu mente

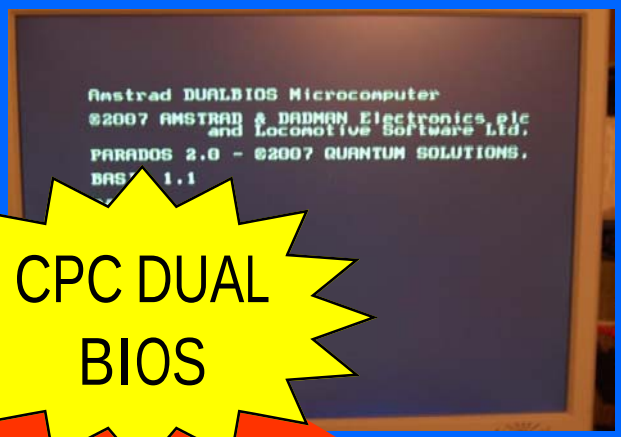
GROOPS!

Curso de Ensamblador (2)

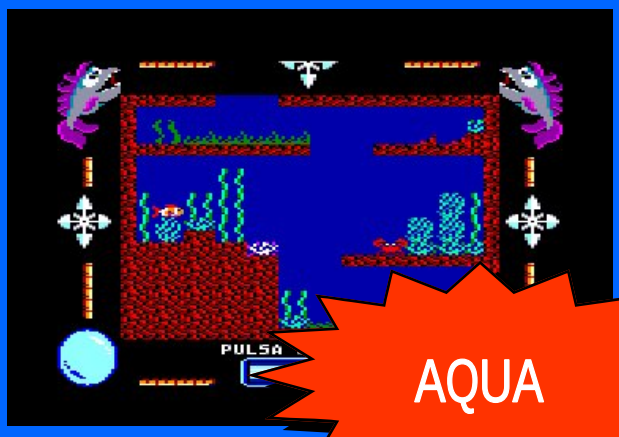
Hablamos con Eliot

Aprendiendo dBASE II

Novedades y Reviews



CPC DUAL BIOS



AQUA

¡ Todas las claves de Groops ! Y, además, previews de juegos en preparación

CEZ GAMES STUDIO, LÍDER EN VIDEO-JUEGOS

¡OBLONGO TE NECESITA!

STRATOS

Stratos es un juego en formato ROM 8Kb para ordenadores MSX de primera generación. Está programado expresamente para el concurso MSXDev'04 y se trata de un rápido arcade de laberintos que combina dos estilos, acción y estrategia.

DISPONIBLE PARA MSX

PANTALLA VERSIÓN MSX

¡VERÁS BLOQUES POR TODOS LADOS!

COLUMNS

Columns es un juego en el que tienes que ir eliminando piedras y ganando puntos, de modo que pulverices el record del mundo de piedras retiradas. Está basado en el original de SEGA, un juego que hizo historia allá por los años noventa e incluso, hoy en día, siguen sacando versiones nuevas con opciones novedosas pero con la idea del original: retirar gemas en bloques de 3 o más.

DISPONIBLE PARA SPECTRUM Y AMSTRAD CPC

PANTALLA VERSIÓN SPECTRUM

¡NO PODRÁS PARAR DE JUGAR!

MOGGY

En el bosque de los canutos, los bolotes de la aldea de Mu, cerca de Kónrad, celebraban una gran fiesta.
Entre el jolgorio, el malvado hechicero Hibön había escogido la mejor ocasión para llevar a cabo sus planes: los bolotes, embriagados y sumisos, no ofrecerían la menor resistencia. Cual flautista de Hamelín, los engatusaría para "seguir la fiesta en otra parte".
Moggy, que había conseguido escapar a los encantos debido a que estaba durmiendo la mona, decidió emprender el rescate de manera inmediata.

DISPONIBLE PARA SPECTRUM

PANTALLA VERSIÓN SPECTRUM

¡SÓLO TÚ PUEDES SALVARNOS!

GATES TO HELL

Penetra en el inespugnable búnker de Microchoft para acabar con la terrible amenaza que se cierne sobre la humanidad, el nuevo Sistema Operativo: Windows to Hell, desarrollado por el malvado Mr. Gates. Se trata de tu misión más difícil..

DISPONIBLE PARA AMSTRAD CPC

PANTALLA VERSIÓN AMSTRAD CPC

¡ESTO SI QUE ES MULTIPLATAFORMA!

PHANTOMAS SAGA: INFINITY

Los últimos acontecimientos habían enseñado a Phantomas que la única forma de atajar un mal era de raíz. Durante siglos, una misteriosa civilización oculta en algún lugar de la nebulosa de Andrómeda, había regido los destinos del Universo de forma encubierta...

DISPONIBLE PARA SPECTRUM, AMSTRAD CPC Y MSX

PANTALLA VERSIÓN AMSTRAD CPC

¡A MACHACAR LADRILLOS!

RAGNABLOCK

23 de Agosto de 2035.
La nave Ragna III, bajo el mando de tus expertas manos, se precipita en el espacio atraída por un agujero de gusano.
Cuando el viaje al hiperespacio finaliza, recuperas el control para observar con horror que te encuentras en un laberinto en medio de un extraño campo de asteroides custodiado por naves alienígenas, y dentro del cual se abren agujeros de gusano que llevan a bloques prisión, de los que tienes que salir destruyendo los bloques con el armamento de tu nave, todavía activo.

DISPONIBLE PARA SPECTRUM

PANTALLA VERSIÓN SPECTRUM

También puedes probar PITFALL ZX, RUN BILL RUN y GALAXY FIGHTER para SPECTRUM y el aclamado por crítica y público SIR FRED REMAKE para PC!!!

¿LOS TIENES TODOS?

¿A QUÉ ESPERAS? DESCÁRGALOS GRATIS DESDE:

cezgs.computeremuzone.com

© 2004-2006 COMPUTER EMUZONE GAMES STUDIO (CEZ GS). Part of the Computer Emuzone network.

En este número

Pag. 4 **Lo Último:** *Ponte al día con todo lo relacionado con el CPC*

Pag. 7 **El Juego:** *Groops*

Pag. 9 **Hoy hablamos con:** *Eliot*

Pag. 12 **Programación:** *Mi primer programa en ensamblador (II)*

Pag. 23 **MiniReviews:** *Recordando clásicos*

Pag 26 **El Taller:** *CPC Dual Bios*

Pag. 35 **El TopTen de…** *Litos*

Pag. 38 **Sacando partido al Amstrad:** *Aprendiendo dBASE II (III)*

Redacción y Colaboradores: Bisarma, Scream, Artaburu, MiguelSky, Litos, DaDMaN, C_PINA666 , transformer

Publicidad: amstrad.mail@gmail.com
Edita: Amstrad ESP
Realización: RUA Press
Distribución: Nación Arcade (www.nacionarcade.net)

Prohibida la reproducción total o parcial de los originales de esta publicación sin autorización por escrito.
No nos hacemos responsables de las opiniones emitidas por nuestros colaboradores y anunciantes.

Contactar: amstrad.mail@gmail.com

Editorial...

Entrando en los rigores del verano, presentamos una nueva edición de RUA. En esta ocasión , queremos agradecer la acogida con que se nos ha recibido y por ello volvemos a la carga con nuevas energías y con la intención de mejorar los contenidos que os ofrecemos.

En este número queremos destacar el enfoque eminentemente práctico que desde aquí intentamos imprimir en esta nuestra (vuestra) publicación, por lo que continuamos con los tutoriales dedicados a de dBASE y Ensamblador. Además, como plato fuerte traemos un nuevo proyecto mediante el cual podremos disponer en nuestros Amstrad de dos BIOS seleccionables.

No obstante, estando en estas fechas no dejaremos de lado la parte lúdica de nuestros CPC. Y es por ello que presentamos el primer análisis en español del juego Groops, la nueva creación de Binary Sciences y con cuyo autor presentamos la entrevista en esta edición.

Y como todo no va a ser novedades, hablamos de los viejos clásicos que en su día nos hicieron pasar tan buenas tardes con unas minireviews con las que volveremos a recordarlos y, quien sabe, quizá os hagan a dedicarles otra oportunidad y rejugarlos o descubrir esas pequeñas joyas que no pudisteis disfrutar en su día.

Y aún hay más cosas que encontraréis en este número que os invito a descubrir. Aprovechamos también para dar la bienvenida a las nuevas incorporaciones de la editorial, con cuya ayuda seguro llegaremos a vosotros con unos contenidos aún mejores.

Esperamos que disfrutéis con este nuevo ejemplar de RUA y paséis un buen verano.

La redacción.

TEXTOS: ARTABURU

EVENTOS

MADRISX07

Como cada año, en Marzo se celebró la tradicional reunión retro por excelencia de España en la que hubo representación de usuarios y máquinas de todos los sistemas. El Amstrad CPC estuvo representado por GUA (Grupo de Usuarios de Amstrad) y asistieron al acto nombres tan reconocidos entre los usuarios como: Deepbf, D-o-S, cpcmaniaco, transformer, cngsoft, MiguelSky, Mochilote, DaDMaN, Rockriver, KaosOverride, Pinace, Anjuel, Artaburu… y seguro que me dejo a alguno. De lo allí acontecido hay en Internet muestras suficientes, desde crónicas escritas y reportajes fotográficos, a videos en YouTube, así que os recomendamos que os paséis por un buscador y os entretengáis leyendo y viendo.
http://www.madrisx.org/

CROCO CHANEL

Esta increíble reunión de usuarios celebrada en Francia reunió, del 4 al 6 de Mayo, a unas 80 personas (casi todos franceses) en la que se puede considerar la mayor reunión actual de usuarios CPC en el mundo. Pasaron un fin de semana de trabajo y experiencias de las que salió alguna demo y un montón de fotografías en la web oficial del encuentro:
http://www.revolog.com/cc4/croco.php

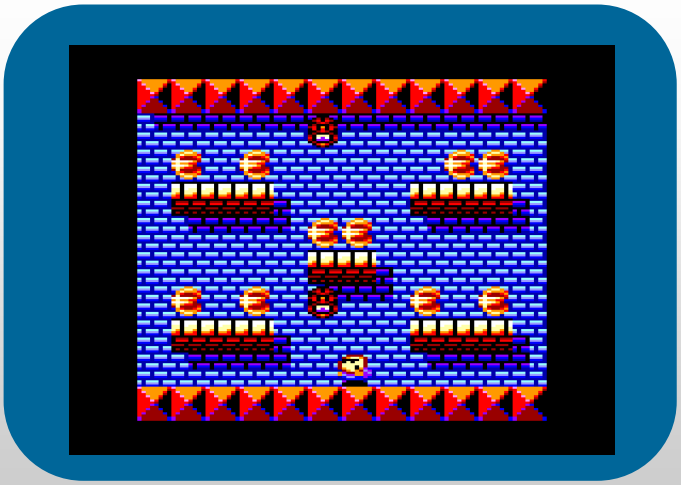
JUEGOS

ELVIRA – MISSTRESS OF THE DARK

***DevilMarkus** ha presentado la conversión del clásico Elvira para CPC. El proyecto se realiza capturando las imágenes del original, y convirtiéndolas al formato CPC. El juego tiene muy buena pinta, y esperamos que no tarde en terminarse y que lo podamos disfrutar también en castellano.*
http://cpc.devilmarkus.de/

UWOL, Quest for Money

Los chicos de CEZ Games Studio están trabajando en un lanzamiento conjunto para todos los sistemas 8bit. Se trata de un juego de plataformas al más puro estilo Bubble Bobble, Mario o Bomb Jack, donde tendrás que guiar a Uwol para coger todas las monedas de un nivel y pasar al siguiente.
http://computeremuzone.com/forum/viewtopic.php?t=3640



Aqua

*Aunque ya presentado meses atrás, este interesante juego que programó **gg** será lanzado por **CEZ GS** en formato casete y disco. Se trata de un juego de plataformas en el que hay guiar hacia la superficie a una burbuja que se encuentra en el fondo del mar, pero hay muchos peligros y no va a ser nada fácil.*

Paciente 106

*Ya lo presentamos como novedad en el número anterior pero, desde entonces, esta aventura conversacional de **Esp Soft** y **CEZ GS Adventures** ha sufrido un lavado de cara muy importante. Para muestra un botón: el aspecto gráfico ha mejorado mucho y el motor ha cambiando también. ¡A ver si en el próximo número ya podemos decir que el juego ha sido publicado!*

SOFTWARE SERIO

SYMBOS

Como novedades sobre este sistema operativo podemos citar la inclusión de un lector de hipertexto que permite navegar por páginas parecidas a las web. Y una versión nueva del SymStudio, software que permite realizar programas para el SymbOS de una forma rápida y sencilla.

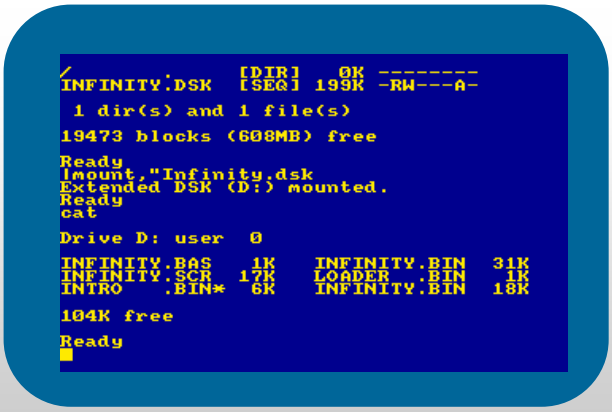
http://www.symbos.de/preview.htm
http://www.symstudio.info

VORTEX TRACKER SOURCE CODE

Se ha liberado el código fuente de este tracker, lo cual es una buena noticia para todos aquellos que quieran adentrarse en las profundidades del programa, bien sea para aprender, o por simple curiosidad.
http://www.kjthacker.f2s.com/download/MusicPlayer.asm

BONNY DOS Y FAT12 DOS

Noob Inc. nos sorprende desde su página Web con diversas utilidades y software, nuevo, e interesante. Por ejemplo, BonnyDOS nos permite usar discos duros desde BASIC y AMSDOS con un sistema de archivos propio (requiere SymbiFace), y con FAT12 DOS (en desarrollo) podemos usar discos de PC/MS-DOS directamente en nuestro CPC.



http://www.geocities.com/timo_brueggmann/bdos.htm

MESCC -M MIKE'S ENHANCED SMALL COMPILER

Aunque para PCW, se ha realizado un compilador C que hará las delicias de los que busquen programar en C para este modelo de Amstrad. El autor: ***Miguel I. Garcia Lopez .***

http://pag-per.servicam.com/migl/amstradpcw

TEXTOS: BISARMA

Aún recuerdo aquellos maravillosos años en los que solíamos jugar, entre otros títulos, a juegos como el Tetris, E-motion, Atomino, Plotting, etc. En la Micromanía eran llamados de Inteligencia o Habilidad. ¿Y qué es Groops?, pues no es más que un juego simple, pero divertido y que se podría clasificar en este grupo de juegos para estrujarse el cerebro.

El objetivo en Groops es algo tan simple, y a la vez tan complicado, como hacer grupos lo más numerosos posibles de iguales elementos. A más elementos combinados más puntos se conseguirán y más alta será nuestra clasificación en el ranking

En Groops existen hasta cinco elementos seleccionables en el menú de inicio a citar y describir:

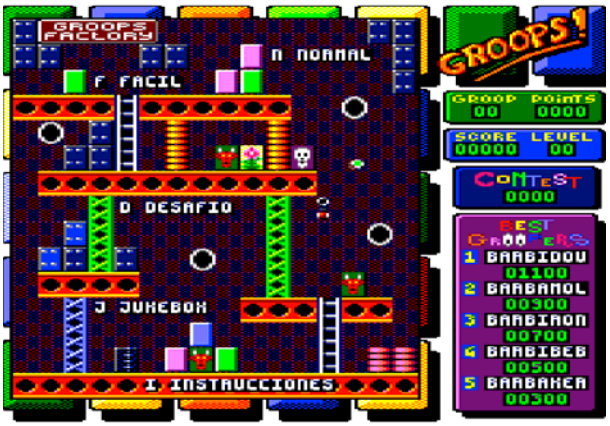
- Modo fácil: El objetivo en esta modalidad de juego es conseguir el mayor numero de puntos con el fin de dejar nuestro récord en los puestos de honor de la tabla. Cabe recordar que el juego posee la capacidad de grabar los records de forma permanente en el diskette.

- Modo normal: exactamente igual que el fácil pero complicando aún más la cosa.

- Instrucciones: desde aquí podremos acceder a las explicaciones de como jugar a Groops. Pero no sólo eso, sino que además dispondremos de una serie de contenidos "extras" en la que se nos cuenta el origen del juego, como surgió la idea, etc.. Imprescindible no perdérselo . Citar que todo está traducido al castellano tras haber seleccionado al principio (antes del menú) el idioma español entre los cuatro posibles: francés, alemán, inglés y castellano.

- Jukebox: nos permite seleccionar y escuchar las melodías incluidas en el juego.

- Desafío: dejo este modo para el final porque creo que es la esencia en sí del propio juego. Lo que hay que hacer aquí es pasar hasta 16 fases de dificultad variable, teniendo que llegar a un número determinado de puntos en cada una de ellas para poder pasar a la siguiente. Tenemos una pequeña ayuda que son unos códigos que se nos facilitarán una vez hayamos pasado una fase y que nos permitirán, otra vez que volvamos a jugar, no empezar desde el principio, sino desde aquella fase en la que lo dejamos.

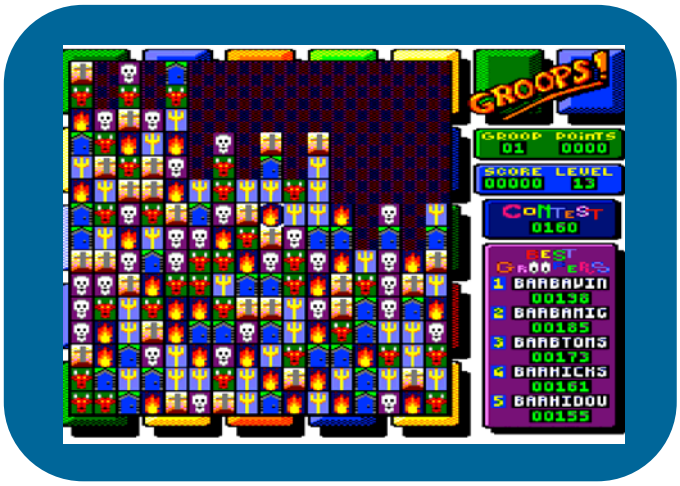


Dicho esto, vamos a pasar a los aspectos técnicos del juego. Decir que Groops en este apartado anda bastante sobrado. Unos gráficos bien realizados, a todo color, y, atención al dato, se ejecuta a pantalla completa, es decir, no tiene los míticos bordes negros por los cuatro laterales de la pantalla que tienen todos los juegos de CPC, o por lo menos la mayoría. A todo ello añadimos que no es para nada lento y una música muy bien realizada, ¿y qué nos sale?, pues uno de los mejores juegos nuevos que podemos disfrutar en nuestros queridos Amstrad CPC.

Ahora sí, que lo que marca la diferencia totalmente es la adicción que crea. Y esto no lo digo por decir, sino que yo hasta que no me lo ventilé no he dejado de jugar. Dadle una oportunidad y veréis que pasada en el modo Desafío.

Como conclusión decir que Binary Sciencies nos ha regalado un título increíble y muy bien realizado a todos los niveles. Simple en su concepción, pero tremendamente adictivo, que al fin y al cabo es lo que cuenta en un juego, que entretenga lo máximo posible, y en eso Groops se lleva la palma.

Y recordad: si os quedáis con ganas de más Binary Sciences, próximamente tendréis en vuestros CPC "Sudoku Master", su próxima creación.



Las claves

Aquí tenéis todos los códigos del modo desafío.

1. WATERZOO	*2. BAZOOKAS*	*3. ANOTHEAW*	*4. BOOLYGAM*
5. SUKAOUTE	*6. OLORAM*	*7. PIZZAHIT*	*8. FASHIONS*
9. ROZOTOOM	*10. MECANISM*	*11. NORMANDE*	*12. HARDEAR*
13. RAINBOWS	*14. OLYMPISM*	*15. PUBISOFT*	

Global ★★★★★★★★★

Hoy hablamos con...

OLIVIER FLOQUET (ELIOT)

POR ARTABURU

En cuanto vimos Groops y lo bien hecho que estaba nos pusimos inmediatamente en contacto con Olivier Floquet (Eliot), el programador del juego, para solicitarle una entrevista en la que nos contara sus experiencias y trucos.

RUA: En primer lugar, muchas felicidades por tu juego, está muy bien hecho y es muy adictivo. ¿Por qué decidiste programar un nuevo juego para el CPC?

*Eliot: Encantado de que disfrutaras jugando a Groops. ¡Groops no es mi primer juego!. Ya lancé dos juegos anteriormente, Solomon's Key 3 en 1994 y Push en 1995, cuando empecé a programar en BASIC ayudado con algunas rutinas para dibujar sprites. Después estuve principalmente interesado en hacer efectos de demo. Tengo algunos efectos preparados para ser presentados pero a la vez decidí hacer un juego, un nuevo desafío para mí porque me dí cuenta de que hay un montón de gente en el mundo más interesada en juegos que en demos. Encontré un concepto de juego interesante en el PC para adaptar al CPC, entonces empecé a programar lo más importante: un dibujado rápido para toda la pantalla de juego (256 bloques, 4 bytes *16 líneas cada bloque). El desafío era tener un juego a pantalla completa, ¡algo extraño en el Amstrad! Le pedí a Barjack/Futur's, un grafista francés con talento, que me hiciera los sprites. Él empezó a trabajar rápidamente pero se sentía un poco limitado con la anchura de los sprites, solo 8 pixels. Entonces, en secreto, comenzó a trabajar con 9 pixeles y pidió a Grim/Arkos que me programara una nueva rutina de dibujo. Una muy buena sorpresa, pero un poco molesta para mí, que había trabajado duramente en mis propias rutinas. Como no podía aceptar no usar mi código decidí buscar otro concepto de juego que adaptar con el código disponible. Hablando con Mig de www.phenixinformatique.com, descubrí un juego escrito en java en una web. ¡Ya había encontrado el concepto! Dibujé unos sprites muy pobres y escribí el núcleo del juego, me refiero a las numerosas comprobaciones que hay que hacer sobre el mapeado. Pedí los gráficos a Slyder ya que había trabajado con él en varias ocasiones y sabía que podía hacer un gran trabajo en Modo 0. Conocí a Napo más tarde durante el proyecto lo cual fue una buena sorpresa.*

¿Quién tuvo la idea del juego?

Como ya expliqué anteriormente, el concepto era conocido en otras plataformas. Pero Groops es una versión específica para CPC con algunas características de nuestra invención. Primero, el nombre lo encontramos entre mi novia Elodie y yo en unos instantes. El modo desafío es idea mía pero Villain sugirió añadir un sistema de contraseñas. Y ya, en los últimos momentos, Slyder propuso añadir una jukebox. Tal vez pensaba que había escrito poco código .

¿Cuánto tiempo has estado programando este juego?

Vamos a decirlo: ¡Mucho tiempo! El hecho es que el juego se empezó a finales de 2004 pero luego se paró. Yo estaba buscando un nuevo concepto, como ya he contado antes. También empecé mi juego Sudoku Master, que debía haber sido lanzado antes que éste, pero como encontramos la idea de Groops, y Slyder empezó con los gráficos, el juego se volvió más importante e incluso ¡más grande de lo que había planeado! Trabajo en varios proyectos a la vez con frecuencia.

Has usado overscan en el juego y realmente queda muy bien, pero esto reduce la memoria disponible, ¿es ésta la razón por la que el juego funcione sólo en un 6128?

¡Esa no es la razón por la que el juego sólo va en un 6128! Siempre programo para CPC 6128, con las características del 6128+ si son útiles. No estoy interesado en hacer juegos o demos para el 464 ya que no conozco a nadie que lo use, excepto los usuarios ingleses que sí que lo utilizan. Los juegos a pantalla completa son escasos en el CPC pero podemos ver alguno como Inferno, Ishido o Xyphoes Fantasy. Por supuesto, un juego no es mejor por usar pantalla completa pero, ¿por qué no usar todas las posibilidades del CPC como la pantalla completa, los rasters, multimodo, etc... cuando es fácil hacerlo? Recuerdo que en el pasado todo el mundo se entusiamaba con las imágenes a pantalla completa de Titux (Crazy Cars, Wild Streets...) o con más de cuatro colores en modo 1.
Técnicamente no hay page-flipping en Groops. Los 240 sprites, todos, se dibujan en 3/50 segundos ¡incluso si sólo hubiera un sprite que cambiar! Así que el tiempo requerido para dibujar siempre es el mismo. Yo dibujo los sprites del siguiente modo: La línea 16 de todos los sprites, luego la 15, la 14,... así que es muy difícil ver la línea de refresco de pantalla. ¡La forma de codificar los sprites es también muy importante para tener una rutina rápida!

¿Qué dificultades tuviste programando el juego? ¿Qué fue lo más difícil?

No encontré grandes dificultades durante la creación. Por supuesto, encontré algunos bugs tal como todo programador puede hacerlo. La mayor dificultad fue mantener la motivación porque fue muy largo de programar. Hay muchas cosas diferentes que hacer y a veces tenía la sensación de que el juego era una historia interminable: Manejo de la puntuación, la rutina para meter el nombre, manejo de contraseñas, grabar, jukebox, instrucciones, niveles aleatorios, pantallas de final, codificar los niveles y por supuesto, una parte oculta. Otra dificultad fue poner todo en un disco ya que tenemos cuatro versiones distintas. El juego comprimido ocupa 58Kbs y añadimos otros 26Kbs por cada idioma, además la intro, el menú de idioma... Al final, tuve que comprimir más los archivos para cambiar la forma de guardar los datos en el disco.

¿Cómo programas? ¿Qué utilidades, ensambladores usas? ¿Eres el único programador del juego?

Yo programo principalmente en el trabajo... ¡No es algo bueno para mi empresa pero sí que es muy bueno para el mundo del CPC! Yo uso el ensamblador de WinAPE. Por supuesto que hice un montón de pruebas en mi CPC ya que el emulador no es totalmente fiable, sobre todo en el acceso a disco, así que también usé Dams y Hack-it para depurar. Slyder usó el Paint Shop Pro para la pantalla de introducción y luego la retocó en el CPC con el Graph'OS, unas utilidades para CPC+ que funcionan a pantalla completa. Todos las animaciones de sprites fueron diseñadas usando el legendario OCP. Napo usó STarKos, una sorpresa para mí ya que las pocas melodías que había oído suyas estaban compuestas con Soundtracker 128. Para comprimir los datos, ninguna sorpresa: Crown Cruncher and CPC-T. Nuestro querido BASIC se usó para codificar los sprites, para generar los niveles, las tablas de puntuaciones, etc.

¿Qué es Binary Sciences? ¿Cuándo se creó y quiénes son sus miembros y sus habilidades?

Binary Sciences es un intento de hacer que la gente con talento trabaje en conjunto para hacer juegos. Aunque aún existen algunos grupos (Arkos, Impact, Dirty Minds, Overlanders, Hard'Os, Benediction, GPA...), rara vez hay un proyecto para todo el equipo. Por otra parte hay un montón de gente que gasta su tiempo haciendo cosas sólo para sí mismo, como Napo, el músico. Para el próximo juego de Binary Sciences, los gráficos son de Ced, un artista sin grupo porque Kill.Or.Die. ya no existe.
Binary Sciences nació durante la creación de Groops y espero que se lancen un montón de juegos con este nombre. ¡Se pueden unir a nosotros artistas y programadores motivados y con ideas!

Dime otros proyectos en los que has trabajado o tienes previstos.

El próximo juego es Sudoku Master, que debería haber sido lanzado antes que Groops. El juego ya está al 85%. Napo está componiendo la música pero los gráficos de Ced ya están listos. Como Groops, el juego será a pantalla completa y usará split-rasters. Nada muy técnico pero hace la pantalla mucho más bonita.

Tengo otro proyecto, un discmag (revista electrónica en disco) llamada No Comment. Creo que el código se empezó en 2004. Tristemente, algunos artículos están obsoletos, como una entrevista a Prodatron. Pero el concepto no es sólo programar una discmag sino un generador de discmags. Con este generador puedo hacer un nuevo número en un disco de datos con nuevos artículos, melodías y gráficos sin tener que cambiar el código. El objetivo es no perder tiempo en modificar la interfaz sino emplearlo sólo en los artículos. La interfaz está programada sólo por mí, pero la discmag será un producto de la escena para la escena.
Pero sigo interesado en hacer demos...

Y, finalmente, ¿cómo ves la escena internacional? ¿Saludable?

He notado una cosa muy extraña pero probablemente buena para el futuro: Existe una escena CPCera en cada territorio histórico con usuarios experimentados y gente nueva... Alemania, España, Francia, Reino Unido e incluso Grecia. Los foros reemplazan a los fancines y revistas que tuvimos durante la época comercial. La gente tiene acceso a los foros, noticias y conocimiento técnico, y pienso que es una fuente de motivación. Es bonito ver cómo emergen proyectos como Elvira - Mistress of Dark de Devilmarkus, Quest For Money de CEZ Game Studio. Aunque a mí me parece que vemos demasiadas previews. Enseñemos menos para dar una gran sorpresa cuando la producción se lance, especialmente para un juego o una demo.
¡Así que aquí estoy, esperando nuevos juegos desde otros países !

El referente hispano en el mundo de los juegos y la emulación

www.nacionarcade.net

POR ARTABURU

En el capítulo anterior hicimos que una pelota rebotara por las paredes. En este número vamos a mover las palas mediante el teclado.

Al igual que en la anterior entrega, usaremos las rutinas que proporciona el firmware, lo cual nos simplificará notablemente la programación.

PONG.

Este sencillo juego consiste en una pelota que rebota en la pantalla y unas palas que controlan los jugadores. El objetivo es devolver la pelota, evitando que se cuele por tu lado, lo que supondría un punto para el adversario.

Diseñando las palas.

Un posibilidad para dibujar nuestras palas es mediante caracteres, al igual que la pelota, y así lo vamos a hacer. Usaremos varias veces el carácter **▌** *uno encima de otro:*

Para mantener el conjunto unido podemos crear una sencilla rutina que será la que lo dibuje. Le indicaremos las coordenadas (del carácter superior) donde se dibujará y la rutina nos lo representará en pantalla.

Como el movimiento será vertical y los tres caracteres siempre se mantienen unidos y a la misma distancia es fácil calcular, de un modo rápido, las otras posiciones.

En el firmware existe una función que es posicionar el cursor en una fila y una columna:

```
TXT SET CURSOR: &BB75
     Entradas: H: Columna L: Fila
```

Y con esto ya podemos hacer nuestra rutina:

```
;FUNCIONES DEL FIRMWARE

txt_set_column equ &BB6F        ;Establece la posición horizontal del cursor
                          ;A tiene la columna
                          ;AF, HL se corrompen

txt_set_row equ &BB72           ;Establece la posición vertical del cursor
                          ;A tiene la fila
                          ;AF, HL se corrompen

txt_wr_char equ &BB5D           ;Escribe un carácter en la posición en la
                          ;se encuentra el cursor de texto.
                          ;A código del carácter que se va a escribir
                          ;AF, BC, DE, HL se corrompen

; Ampliamos las funciones del firmware que ya definimos anteriormente

txt_set_cursor equ &BB75        ;Establece la posición del cursor
                          ;H tiene la columna y L la fila
                          ;AF, HL se corrompen
```

```
rut_dibujar_pala
      ;Entradas: H: Columna, L: Fila
      ;Se corrompen HL y AF
      PUSH HL              ; vamos a usar más adelante los valores originales
      CALL txt_set_cursor
      LD A,143             ;Es el carácter que usamos para dibujar la pala
      CALL txt_wr_char     ;Hemos dibujado nuestro primer carácter
      POP HL
      INC L                ;El siguiente carácter va en la siguiente fila
      PUSH HL
      CALL txt_set_cursor
      LD A,143
      CALL txt_wr_char     ;Hemos dibujado nuestro segundo carácter
      POP HL
      INC L
      CALL txt_set_cursor
      LD A,143
      CALL txt_wr_char     ;Hemos dibujado nuestro último carácter
      RET
```

Esta rutina tan sencilla nos sirve para dibujar todas las palas que queramos en cualquier posición de nuestra pantalla, pero claro, si no borramos la anterior, cuando movamos la pala, dejaremos un rastro.

Una solución sencilla es borrar en la coordenada anterior y luego dibujar en la nueva.
Podríamos aprovechar la rutina anterior para borrar añadiendo un nuevo parámetro: El carácter. Si queremos dibujar usamos el 143 y si queremos borrar el 32. Para guardar el carácter usaremos una variable temporal (caracter_tmp) y así nos ahorramos muchos PUSH y POP:

```
rut_dibujar_caracter
      ;Entradas: H: Columna, L: Fila
      ;A: carácter a dibujar
      ;Se corrompen HL y AF
      LD (caracter_tmp),A        ;Almacenamos el carácter a representar
      PUSH HL                    ;vamos a usar más adelante los valores originales
      CALL txt_set_cursor        ;Posicionamos el cursor
      LD A,(caracter_tmp)        ;Recuperamos nuestro caracter
      CALL txt_wr_char           ;Hemos dibujado nuestro primer carácter
      POP HL
      INC L                      ;El siguiente carácter va en la siguiente fila
      PUSH HL
      CALL txt_set_cursor        ;Posicionamos el cursor
      LD A,(caracter_tmp)        ;Recuperamos nuestro caracter
      CALL txt_wr_char           ;Hemos dibujado nuestro segundo carácter
      POP HL
      INC L
      CALL txt_set_cursor        ;Posicionamos el cursor
      LD A,(caracter_tmp)        ;Recuperamos nuestro caracter
      CALL txt_wr_char           ;Hemos dibujado nuestro último carácter
      RET
      caracter_tmp db 0
```

Esta es la rutina que usaremos para dibujar y borrar las palas.

Detectando el teclado

El firmware proporciona rutinas que nos van a ser válidas para ver si cierta tecla está pulsada. Por ejemplo, para mover la pala de la izquierda lo haremos con QA y la de la derecha con OL.
`KM TEST KEY: &BB1E` *nos va a decir si el carácter en A está pulsado o no. Si está pulsado entonces Z es falso. Ojo, en A va el número de carácter, no el código ASCII (ver esquema más adelante).*
Nuestros números son: Q=67, A=69, O=26, L=28

Antes de nada, vamos a usar 2 variables para indicar la coordenada Y de cada pala y otras 2 para la coordenada Y anterior, para poder borrar la posición previa y no dejar rastro.

```
pala1_y  db 0
pala1_ya db 0
pala2_y  db 0
pala2_ya db 0
```

Entonces, según veamos que se pulsa la tecla de subir o bajar, modificaremos la variable correspondiente.

```
rut_deteccion_teclado
     ;pala 1
_tecla_arriba_1
     LD A,67              ;tecla arriba
     call km_test_key
     JP Z, _tecla_abajo_1 ;si la tecla arriba no se ha pulsado
                          ;miramos la tecla abajo
     LD A,(pala1_y)
     LD (pala1_ya),A      ;guardamos la coordenada actual en la anterior
     DEC A                ;y subimos una posición
     LD (pala1_y),A
     JP _tecla_arriba_2   ; vamos a ver la pala 2.

_tecla_abajo_1
     LD A,69              ;tecla abajo
     call km_test_key
     JP Z, _tecla_arriba_2     ;si la tecla abajo no se ha pulsado
                          ;miramos la tecla arriba de la pala 2
     LD A,(pala1_y)
     LD (pala1_ya),A      ;guardamos la coordenada actual en la anterior
     INC A                ;y bajamos una posición
     LD (pala1_y),A
                          ; vamos a ver la pala 2.


_tecla_arriba_2
     LD A,26              ;tecla arriba
     call km_test_key
     JP Z, _tecla_abajo_2 ;si la tecla arriba no se ha pulsado
                          ;miramos la tecla abajo
     LD A,(pala2_y)
     LD (pala2_ya),A      ;guardamos la coordenada actual en la anterior
     DEC A                ;y subimos una posición
     LD (pala2_y),A
     RET                  ;terminamos la exploración del teclado

_tecla_abajo_2
     LD A,28              ;tecla abajo
     call km_test_key
     RET Z                ;si la tecla abajo no se ha pulsado  terminamos
                          ;la exploración del teclado
     LD A,(pala2_y)
     LD (pala2_ya),A      ;guardamos la coordenada actual en la anterior
     INC A                ;y bajamos una posición
     LD (pala2_y),A
     RET                  ;terminamos
```

Ahora ya tenemos una rutina básica para detectar el teclado y actualizar las coordenadas de las palas. Sólo tiene un pequeño inconveniente. Hay que comprobar que no hayamos llegado a ninguno de los límites (superior o inferior), para ello, antes de actualizar la coordenada se debe comprobar si estamos en un tope, en ese caso no se actualiza nada.

```
rut_deteccion_teclado_v2
      ;pala 1
_tecla_arriba_1
      LD A,67               ;tecla arriba
      call km_test_key
      JP Z, _tecla_abajo_1 ;si la tecla arriba no se ha pulsado
                            ;miramos la tecla abajo
      LD A,(pala1_y)
                            ;antes de guardar la coordenada comprobamos si
                            ;podemos seguir subiendo
      CP 7                  ;limite superior
      JR Z, _tecla_arriba_2      ;si estamos en el límite no hacemos nada y segimos
                            ;con la pala 2

      LD (pala1_ya),A       ;guardamos la coordenada actual en la anterior
      DEC A                 ;y subimos una posición
      LD (pala1_y),A
      JP _tecla_arriba_2         ; vamos a ver la pala 2.

_tecla_abajo_1
      LD A,69               ;tecla abajo
      call km_test_key
      JP Z, _tecla_arriba_2      ;si la tecla abajo no se ha pulsado
                            ;miramos la tecla arriba de la pala 2
      LD A,(pala1_y)

      CP 22                 ;limite inferior
      JR Z, _tecla_arriba_2      ;si estamos en el límite no hacemos nada y segimos
                            ;con la pala 2

      LD (pala1_ya),A       ;guardamos la coordenada actual en la anterior
      INC A                 ;y bajamos una posición
      LD (pala1_y),A
                            ; vamos a ver la pala 2.


_tecla_arriba_2
      LD A,26               ;tecla arriba
      call km_test_key
      JP Z, _tecla_abajo_2 ;si la tecla arriba no se ha pulsado
                            ;miramos la tecla abajo
      LD A,(pala2_y)
      CP 7                  ;limite superior
      RET Z                 ;si estamos en el límite no hacemos nada y salimos


      LD (pala2_ya),A       ;guardamos la coordenada actual en la anterior
      DEC A                 ;y subimos una posición
      LD (pala2_y),A
      RET                   ;terminamos la exploración del teclado

_tecla_abajo_2
      LD A,28               ;tecla abajo
      call km_test_key
      RET Z                 ;si la tecla abajo no se ha pulsado  terminamos
                            ;la exploración del teclado
      LD A,(pala2_y)
      CP 22                 ;limite inferior
      RET Z                 ;si estamos en el límite no hacemos nada y salimos

      LD (pala2_ya),A       ;guardamos la coordenada actual en la anterior
      INC A                 ;y bajamos una posición
      LD (pala2_y),A
      RET                   ;terminamos
```

¡Ya tenemos nuestro código!

Ahora lo mezclamos todo, añadimos un poco de sal y terminado:

```
;Inicializamos las coordenadas de las palas

      LD A,10
      LD (pala1_y),A
      LD (pala1_ya),A
      LD (pala2_y),A
      LD (pala2_ya),A
;dibujamos las palas

      LD A,143
      LD H,2
      LD L,10
      CALL rut_dibujar_caracter

      LD A,143
      LD H,38
      LD L,10
      CALL rut_dibujar_caracter

;Hacemos el bucle principal
bucle_principal
      ;1. Detectamos el teclado
      CALL rut_deteccion_teclado_v2
      ;2. Borramos y dibujamos las palas
      LD H,2
      LD A,(pala1_ya)
      LD L,A
      LD A,32
      CALL rut_dibujar_caracter
      LD H,2
      LD A,(pala1_y)
      LD L,A
      LD A,143
      CALL rut_dibujar_caracter

      LD H,38
      LD A,(pala2_ya)
      LD L,A
      LD A,32
      CALL rut_dibujar_caracter
      LD H,38
      LD A,(pala2_y)
      LD L,A
      LD A,143
      CALL rut_dibujar_caracter
      JP bucle_principal   ;vuelta a empezar
```

Este sería nuestro programa. Obviamente es muy mejorable, por ejemplo, que no dibuje y redibuje todo el rato, sólo si se ha movido la pala. Para esto podemos usar una variable que indique si se mueve o no la pala. Con una variable podemos controlar las dos palas usando bits en lugar del byte.
Definimos nuestra variable:

```
control_pala db 0
```

Si movemos la pala 1 usamos el bit 0 y lo ponemos a 1 con

```
      LD HL,control_pala
      SET 0,(HL)
```

y si movemos la pala 2 usamos el bit 1 y lo ponemos a 1 con

```
      LD HL,control_pala
      SET 1,(HL)
```

Y al terminar el ciclo lo reseteamos.

```
      XOR A
      LD (control_pala),A
```

Entonces tendremos una nueva rutina para detectar el teclado:

```
rut_deteccion_teclado_v3
      ;pala 1
_tecla_arriba_1
      LD A,67               ;tecla arriba
      call km_test_key
      JP Z, _tecla_abajo_1 ;si la tecla arriba no se ha pulsado
                            ;miramos la tecla abajo
      LD A,(pala1_y)
                            ;antes de guardar la coordenada comprobamos si
                            ;podemos seguir subiendo
      CP 7                  ;limite superior
      JR Z, _tecla_arriba_2      ;si estamos en el límite no hacemos nada y segimos
                            ;con la pala 2

      LD (pala1_ya),A       ;guardamos la coordenada actual en la anterior
      DEC A                 ;y subimos una posición
      LD (pala1_y),A

      LD HL,control_pala
      SET 0,(HL)
                            ;Indicamos que hemos movido la pala
      JP _tecla_arriba_2         ;vamos a ver la pala 2.

_tecla_abajo_1
      LD A,69               ;tecla abajo
      call km_test_key
      JP Z, _tecla_arriba_2      ;si la tecla abajo no se ha pulsado
                            ;miramos la tecla arriba de la pala 2
      LD A,(pala1_y)

      CP 22                 ;limite inferior
      JR Z, _tecla_arriba_2      ;si estamos en el límite no hacemos nada y segimos
                            ;con la pala 2

      LD (pala1_ya),A       ;guardamos la coordenada actual en la anterior
      INC A                 ;y bajamos una posición
      LD (pala1_y),A

      LD HL,control_pala         ;Indicamos que hemos movido la pala
      SET 0,(HL)

                            ;vamos a ver la pala 2.

_tecla_arriba_2
      LD A,26               ;tecla arriba
      call km_test_key
      JP Z, _tecla_abajo_2 ;si la tecla arriba no se ha pulsado
                            ;miramos la tecla abajo
      LD A,(pala2_y)
      CP 7                  ;limite superior
      RET Z                 ;si estamos en el límite no hacemos nada y salimos


      LD (pala2_ya),A       ;guardamos la coordenada actual en la anterior
      DEC A                 ;y subimos una posición
      LD (pala2_y),A
      LD HL,control_pala         ;Indicamos que hemos movido la pala
      SET 1,(HL)
      RET                   ;terminamos la exploración del teclado
```

```
_tecla_abajo_2
     LD A,28                ;tecla abajo
     call km_test_key
     RET Z                  ;si la tecla abajo no se ha pulsado  terminamos
                            ;la exploración del teclado
     LD A,(pala2_y)
     CP 22                  ;limite inferior
     RET Z                  ;si estamos en el límite no hacemos nada y salimos

     LD (pala2_ya),A        ;guardamos la coordenada actual en la anterior
     INC A                  ;y bajamos una posición
     LD (pala2_y),A
     LD HL,control_pala         ;Indicamos que hemos movido la pala
     SET 1,(HL)
     RET                    ;terminamos
```

Y nos queda nuestro programa completo:

```
_inicio código_
;FUNCIONES DEL FIRMWARE

txt_set_column equ &BB6F        ;Establece la posición horizontal del cursor
                            ;A tiene la columna
                            ;AF, HL se corrompen

txt_set_row equ &BB72           ;Establece la posición vertical del cursor
                            ;A tiene la fila
                            ;AF, HL se corrompen

txt_wr_char equ &BB5D           ;Escribe un carácter en la posición en la
                            ;se encuentra el cursor de texto.
                            ;A código del carácter que se va a escribir
                            ;AF, BC, DE, HL se corrompen

; Ampliamos las funciones del firmware que ya definimos anteriormente

txt_set_cursor equ &BB75        ;Establece la posición del cursor
                            ;H tiene la columna y L la fila
                            ;AF, HL se corrompen
km_test_key equ &BB1E


org &4000

;Inicializamos las coordenadas de las palas
     LD A,10
     LD (pala1_y),A
     LD (pala1_ya),A
     LD (pala2_y),A
     LD (pala2_ya),A

;dibujamos las palas en su posición inicial
     LD A,143
     LD H,2
     LD L,10
     CALL rut_dibujar_caracter

     LD A,143
     LD H,38
     LD L,10
     CALL rut_dibujar_caracter
```

```
;Hacemos el bucle principal
bucle_principal

      ;0. Inicializamos los controles de las palas
      XOR A                ;Equivale a LD A,0 pero más rápido
      LD (control_pala),A

      ;1. Detectamos el teclado
      CALL rut_deteccion_teclado_v3

      ;2. Borramos y dibujamos las palas solo si se han movido
      LD HL,control_pala
      BIT 0,(HL)
      JP Z, _mirar_pala2        ;si no se ha movido voy a la pala 2
      LD H,2
      LD A,(pala1_ya)
      LD L,A
      LD A,32
      CALL rut_dibujar_caracter
      LD H,2
      LD A,(pala1_y)
      LD L,A
      LD A,143
      CALL rut_dibujar_caracter

_mirar_pala2
      LD HL,control_pala
      BIT 1,(HL)
      JP Z, bucle_principal          ;si no se ha movido vuelve a empezar

      LD H,38
      LD A,(pala2_ya)
      LD L,A
      LD A,32
      CALL rut_dibujar_caracter
      LD H,38
      LD A,(pala2_y)
      LD L,A
      LD A,143
      CALL rut_dibujar_caracter
      JP bucle_principal   ;vuelta a empezar

rut_deteccion_teclado_v3
      ;pala 1
_tecla_arriba_1
      LD A,67              ;tecla arriba
      call km_test_key
      JP Z, _tecla_abajo_1 ;si la tecla arriba no se ha pulsado
                           ;miramos la tecla abajo
      LD A,(pala1_y)
                           ;antes de guardar la coordenada comprobamos si
                           ;podemos seguir subiendo
      CP 7                 ;limite superior
      JR Z, _tecla_arriba_2     ;si estamos en el límite no hacemos nada y segimos
                           ;con la pala 2

      LD (pala1_ya),A      ;guardamos la coordenada actual en la anterior
      DEC A                ;y subimos una posición
      LD (pala1_y),A

      LD HL,control_pala
      SET 0,(HL)
                           ;Indicamos que hemos movido la pala
      JP _tecla_arriba_2        ;vamos a ver la pala 2.
```

```
_tecla_abajo_1
     LD A,69              ;tecla abajo
     call km_test_key
     JP Z, _tecla_arriba_2     ;si la tecla abajo no se ha pulsado
                          ;miramos la tecla arriba de la pala 2
     LD A,(pala1_y)

     CP 22                ;limite inferior
     JR Z, _tecla_arriba_2     ;si estamos en el límite no hacemos nada y segimos
                          ;con la pala 2

     LD (pala1_ya),A      ;guardamos la coordenada actual en la anterior
     INC A                ;y bajamos una posición
     LD (pala1_y),A

     LD HL,control_pala        ;Indicamos que hemos movido la pala
     SET 0,(HL)

                          ;vamos a ver la pala 2.


_tecla_arriba_2
     LD A,26              ;tecla arriba
     call km_test_key
     JP Z, _tecla_abajo_2 ;si la tecla arriba no se ha pulsado
                          ;miramos la tecla abajo
     LD A,(pala2_y)
     CP 7                 ;limite superior
     RET Z                ;si estamos en el límite no hacemos nada y salimos


     LD (pala2_ya),A      ;guardamos la coordenada actual en la anterior
     DEC A                ;y subimos una posición
     LD (pala2_y),A
     LD HL,control_pala        ;Indicamos que hemos movido la pala
     SET 1,(HL)
     RET                  ;terminamos la exploración del teclado


_tecla_abajo_2
     LD A,28              ;tecla abajo
     call km_test_key
     RET Z                ;si la tecla abajo no se ha pulsado  terminamos
                          ;la exploración del teclado
     LD A,(pala2_y)
     CP 22                ;limite inferior
     RET Z                ;si estamos en el límite no hacemos nada y salimos

     LD (pala2_ya),A      ;guardamos la coordenada actual en la anterior
     INC A                ;y bajamos una posición
     LD (pala2_y),A
     LD HL,control_pala        ;Indicamos que hemos movido la pala
     SET 1,(HL)
     RET                  ;terminamos



control_pala db 0
pala1_y  db 0
pala1_ya db 0
pala2_y  db 0
pala2_ya db 0
```

```
rut_dibujar_caracter

     ;Se corrompen HL y AF
     LD (caracter_tmp),A        ;Almacenamos el carácter a representar
     PUSH HL               ;vamos a usar más adelante los valores originales
     CALL txt_set_cursor        ;Posicionamos el cursor
     LD A,(caracter_tmp)   ;Recuperamos nuestro caracter
     CALL txt_wr_char           ;Hemos dibujado nuestro primer carácter
     POP HL
     INC L                 ;El siguiente carácter va en la siguiente fila
     PUSH HL
     CALL txt_set_cursor        ;Posicionamos el cursor
     LD A,(caracter_tmp)   ;Recuperamos nuestro caracter
     CALL txt_wr_char           ;Hemos dibujado nuestro segundo carácter
     POP HL
     INC L
     CALL txt_set_cursor        ;Posicionamos el cursor
     LD A,(caracter_tmp)   ;Recuperamos nuestro caracter
     CALL txt_wr_char           ;Hemos dibujado nuestro último carácter
     RET
     caracter_tmp db 0

_fin código_
```

Para ejecutarlo podéis copiar el código en el MAXAM de WinAape, compilarlo y ejecutarlo con call &4000.

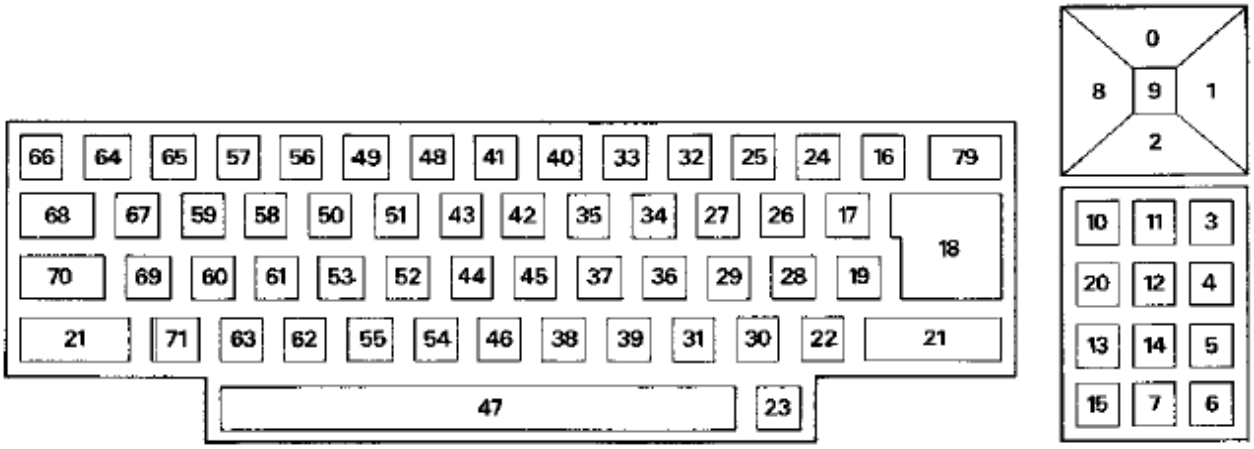
instrucciones nuevas ASM utilizadas en este programa:

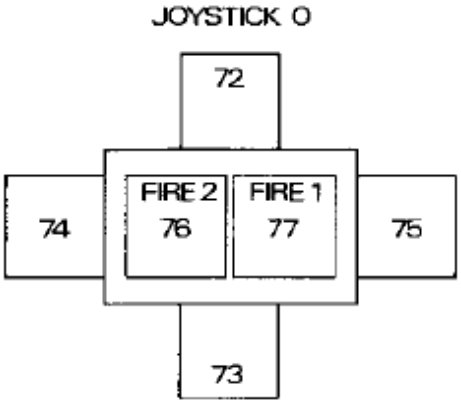
XOR A
Realiza la operación XOR entre A y A. Cuando dos bits son iguales pone 0 y si son distintos pone 1.
XOR (HL) hace la oparación entre A y (HL)

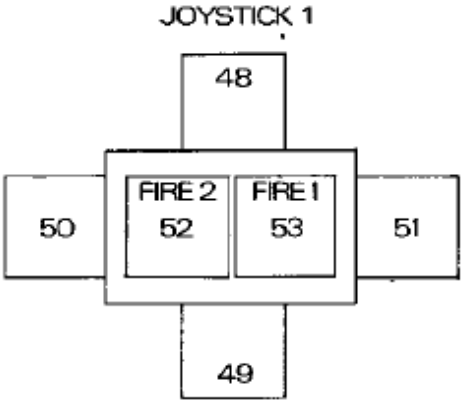
SET bit,(HL)
SET bit,A
pone a 1 el bit indicado de A o del byte que está en la dirección que apunta HL.

BIT bit,A
BIT bit,(HL)
mira el bit de A o del byte que está en la dirección que apunta HL y actualiza Z en función del resultado.

Esquema de los valores de las teclas y de los joysticks (Cortesía del manual del CPC664)







TEXTOS: |C_PINA666

MATCH DAY 2

Puntuación Global

***Compañía**: Ocean*

***Genero:**Fútbol*

Bueno, que menos que comenzar mi andadura como colaborador de esta carismática revista que hacerlo con un juego de fútbol, deporte rey por estos lares. Este clásico de Jon Ritman y Bernie Drummond supuso una buena piedra de toque para los juegos futbolísticos de los 8 bits de finales de los 80.

Este juego es la segunda parte de un exitoso Match Day en el cual su sistema de juego es simple pero a la vez muy completo, de los mejores simuladores deportivos de la época.

Vamos por partes:

Jugabilidad: el control al principio puede hacerse un poco complicado pero, en cuanto te haces al balón y controlas la barra de fuerza para la patada, puedes llegar a enlazar pases y hasta centros y remates de cabeza muy espectaculares.

Bueno voy a jugar un Ritman Utd vs Ocean Blue…

Gráficos: aunque ahora en el 2007 parezcan unos Shin-Shan futbolistas, en su momento fueron gráficos considerables y las palomitas de los porteros son geniales.

Sonido: el sonido también era bueno para su época (la cancioncita de antes de empezar el partido es superpegadiza).

Adicción: el sistema de liga y de copa hace que tuvieras para entretenerte horas y horas, y el sistema de claves para poder continuar cuando quisieras no lo hacia pesado si querías completar la liga.

***En resumen:** de los mejores títulos de fútbol para nuestro CPC y en general para los añorados 8 bits.*

***Puntuación:** 8,5/10* ***La puntuación de la redacción:** 8,5/10*

ARKANOID

Puntuación Global

Compañia:*Taito*

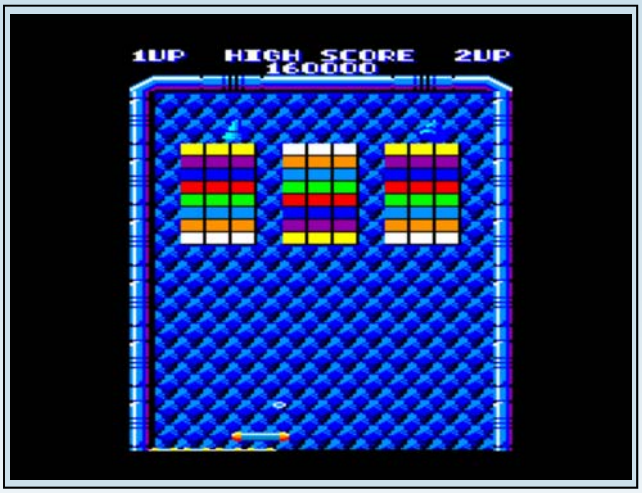
Genero: *Arcade*

Situémonos, la nave nodriza Arkanoid ha sido destruida, una pequeña nave llamad Vaus pudo escapar pero en su huida entró en un universo corrompido por alguien... La verdad que darle argumento a este tipo de juegos tiene su habilidad.

Este clásico de clásicos es una adaptación de una recreativa con bastante éxito, quedando un arcade de lo mas resultón. Este juego original de 1987 resulto un gran éxito y se convirtió en uno de un, como dicen ahora los hardcore gamers (llámese nueva generación de viciados), "must have" de los 80.

Vamos por partes

Jugabilidad: el juego es muy simple, hay que romper, con una bolita que rebota por todas partes, los ladrillos evitando que caiga al vacío. Para eso está nuestra nave con forma de plataforma que moveremos a un lado y a otro. Entre los ladrillos habrá distintos marcianitos que nos harán la puñeta estorbando la mayoría de veces. Cuando rompemos lo ladrillos nos pueden caer distintos ítems que tendrán diversos efectos sobre nuestra nave, véase alargarla, hacer que la bolita se quede pegada, dividir la bolita en tres, etc...



Gráficos: aunque son simples, el colorido tan alegre y los suaves movimientos hacen un acabado muy vistoso.

Sonido: las melodías están cuidadas y suenan muy futuristas. Bien en general.

Adicción: aunque es un juego difícil, resulta muy adictivo. Quizás este es su punto fuerte en todo un global, porque aunque tiene una mecánica simple, engancha bastante. Ahora un consejo si te lo quieres terminar: o te haces un maestro, o prepárate un cargador de vidas infinitas.

En resumen: *un arcade clásico con la única pega de su excesiva dificultad.*

Puntuación: *8,75/10* ***La puntuación de la redacción:*** *7,5/10*

GONZZALEZZ

Compañía: Opera Soft

Puntuación Global

Genero: Plataformas/Videoaventura

Una de las mejores compañías de soft española de la época de los 8 bits fue sin duda Opera Soft. Y uno de sus juegos más curiosos fue este Gonzzalezz.

Nuestro manito Gonzzalezz es un hombre de lo más dormilón (que cliché más típico de los mejicanos, ¿eh?) y nuestro objetivo es ayudarlo, primero, a que llegue a buen puerto en su pesadilla para que pueda apagar el despertador y descansar tranquilamente, y segundo, a alcanzar la hamaca atravesando el desierto y ciertos parajes del oeste americano fronterizo con México. Todo para que nuestro bigotudo amigo descanse tranquilamente a gusto.

El juego, como muchos de su época, está dividido en dos partes que bien podían ser dos juegos distintos.

La primera parte es totalmente de plataformas en las que deberemos pasar saltando a través de 7 fases enlazadas entre sí para completarla. Cada fase tendrá su paisaje propio y sus obstáculos característicos como chorros de agua, imanes que te desestabilizan, aspiradoras, etc...

Una vez completada, la segunda parte se trata de una videoaventura en la que deberemos atravesar el desierto y enfrentarnos a pieles rojas, bichos y demás, además de coger determinados objetos y usarlos adecuadament e.

Vamos por partes:

Jugabilidad: en la primera parte deberemos saltar de plataforma en plataforma pudiendo controlar la dirección del salto en todo momento, de un modo muy suave y muy conseguido.

Quizás sea en la segunda parte donde costará hacerse más al control debido a la mayor variedad de posibilidades, sobre todo para coger y soltar objetos si te agachas.

Gráficos: bastante conseguidos en ambas partes, con unos colores vistosos y bonitos. Buenos en líneas generales.

Sonido: cancioncitas pegadizas:nanas, rancheras, mortuorias,...y buenos efectos.

Adicción: ya sabemos que hablar de un programa español es hablar de un programa sumamente difícil, y éste no es una excepción, pero para los jugones supondrá todo un reto al tratarse de un juego bastante divertido, que siempre pide repetir. Muy adictivo.

En resumen: un programa típico del soft patrio, con sus dos partes totalmente distintas, de elevada dificultad y tremenda calidad que demostraba lo bueno que era en los 8 bits. Por cierto destacar la genial portada de Azpiri.

Puntuación: 9/10 **La puntuación de la redacción:** 6,5/10

TEXTOS: DADMAN

¿Cuántos de vosotros habéis deseado alguna vez tener vuestro querido CPC "tuneao"? Cambiar el tipo de letra predeterminado al arrancarlo, los colores, formatear discos a 800Kbytes… Todo esto es posible modificando la ROM y sustituyendo la original del CPC por nuestra versión personalizada. Pero claro, ¿Qué pasa si queremos volver al estado original? ¿Tenemos que abrir nuestro ordenador cada vez para intercambiar las ROMs por un problema de incompatibilidad?

Para solventar ese pequeño problema y que pueda llover a gusto de todos, vamos a dotar a nuestro CPC de 2 sistemas ROM independientes seleccionables a nuestro gusto o disgusto; un "doble arranque". ¡EMPECEMOS!

UNOS CUANTOS DATOS A TENER EN CUENTA

Vamos a empezar explicando un poco lo que pasa cuando encendemos el Amstrad CPC.

Cuando damos "candela" al equipo lo primero que ocurre es que vemos una maravillosa pantalla de color azul y unas letras amarillas que nos dan cierta información, como el año del Copyright, versión de BASIC y un maravilloso mensaje "Ready" que nos indica que el sistema ya está listo para ser usado. Todo esto son programas "precargados" en la memoria del ordenador y que no se borran al apagarlo. Es lo que se denominan programas en ROM (Read Only Memory), o lo que es lo mismo: Memoria de Sólo Lectura.

Lo que nosotros vamos a hacer es crearnos unas memorias ROM personalizadas para sustituirlas por las originales. Concretamente nos vamos a centrar en el Amstrad CPC 6128 y los programas ROM que vamos a sustituir son los siguientes:

OS (ROM baja): Esta es la primera ROM que el equipo lee al ponerse en marcha. Tiene las llamadas al firmware y es la que muestra el mensajito "Amstrad 128K Microcomputer…" entre otras muchísimas cosas. Su tamaño es de 16Kbytes (16384 bytes).
BASIC (ROM del intérprete BASIC): Esta es la segunda ROM y contiene el intérprete BASIC (en nuestro caso, la versión 1.1) y junto a la anterior ROM es la que nos va a permitir comunicarnos con el ordenador tras su puesta en marcha. Su tamaño es de 16Kbytes (16384 bytes).
AMSDOS (ROM de disco): Esta es la tercera y última ROM que viene instalada de serie en los Amstrad CPC6128 (y el CPC664 también). Es la encargada de que podamos leer y escribir en los disquetes y CATalogar el contenido de un disco desde el BASIC. Los Amstrad CPC 464 carecen de esta ROM ya que no vienen con unidad de disco incorporada de serie. Su tamaño es de 16Kbytes (16384 bytes).

Estos programas ROM están almacenados en 2 chips (memorias). El primero de ellos es el 40025 para la versión inglesa (40038 para la versión española y 40051 para la versión francesa) y tiene una capacidad de 32Kbytes, conteniendo el OS y el BASIC. El segundo es el 40015 y es donde se almacena el AMSDOS (o ROM de disco) y tiene una capacidad de 16Kbytes.
Como lo que nosotros queremos es cambiar estas ROMs por unas personalizadas pero teniendo la posibilidad de volver a las originales sin abrir el ordenador, vamos a necesitar unas memorias con el doble de capacidad que las originales, y aquí entramos en la lista de…

COMPONENTES

***EPROM 27C512**: Esta memoria tiene una capacidad de 64Kbytes y su precio ronda los 6 €. Es justo lo que necesitamos para almacenar el OS y el BASIC originales (32Kbytes) y el OS y el BASIC modificados (32Kbytes). Esto hacen un total de 64Kbytes.*
***EPROM 27C256:** Esta memoria tiene una capacidad de 32Kbytes y su precio ronda los 5 €. En ella almacenaremos el AMSDOS original y el PARADOS (sustituto del AMSDOS con muchas nuevas funcionalidades y "casi" compatible 100% con el AMSDOS). Cada una ocupa 16Kbytes, 32Kbytes en total sumando ambas.*
2 zócalos DIP de 28 pines para pinchar nuestras ROMS. Sobre 1 € (o menos) cada uno.
Una ristra de espadines para placa impresa (esos donde se pinchan los jumpers de toda la vida). Si le caes bien al de la tienda, tal vez te los regala. En el peor de los casos 0.50 € por una hilera de 25 espadines. Podéis cambiar los espadines por un conmutador para chasis de 2 posiciones.
Malla para eliminar soldaduras. La emplearemos para quitar el estaño de las soldaduras de las ROMs originales en la placa base del ordenador.
Cablecillos varios, estaño, soldador de punta fina y un poco de paciencia.
Opcionalmente podéis comprar un "ayudante" para la soldadura: JBC FLUX 9600. Es un botecito de 15ml que nos ayudará a eliminar las antiguas soldaduras junto a la malla y para fijar mejor las nuevas que hagamos (aunque no es imprescindible).
Archivos de ROM originales y los modificados por vosotros (o por un colega) para meterlos en vuestro flamante CPC.
Alguien que os programe las EPROMs anteriormente citadas (o vosotros mismos si tenéis programador de EPROMS). Si queréis comprar uno, rondan los 300 €.

Con todo este material en nuestras manos, ya estamos preparados para empezar a…

METERLE MANO A LAS ROMS ORIGINALES

Ya se que suena un poco cochino, así que voy a explicarme. Lo que vamos a hacer es extraer las ROMS originales de la placa base y para posteriormente poder soldar en su lugar los zócalos donde irán pinchadas nuestras nuevas ROMS.

Para ello, abrimos el ordenador (sí, hay que quitar todos los tornillos) y tras desenchufar todos los cables que van a la placa base, quitamos toda su tornillería (4 tornillos) y la separamos de la carcasa de plástico. Ahora podemos ponerla encima de la mesa boca abajo y con el soldador calentito en una mano y la malla en la otra empezar a desoldar.

Hay varios métodos para conseguir quitar las soldaduras. Yo, concretamente he seguido el siguiente:

Primero humedezco la soldadura en cuestión con el FLUX (liquidito maravilloso).
Seguidamente coloco el extremo de la malla sobre la soldadura
Coloco el soldador encima de la malla para calentarla
Espero a que la malla empiece a chupar el estaño (se ve claramente cuando cambia de color cobre a color "plata", señal de que el estaño se ha transferido a la malla).
Retiro la malla y empezamos con la siguiente soldadura…

Aunque pueda parecer engorroso, no lo es en absoluto. Tan sólo hay que tener un poco de paciencia y tesón. Yo tardé una media hora por cada chip. IMPORTANTE recordar que la placa base del CPC es de doble cara y que hay soldadura en ambos lados de los CHIPS. De todos modos, con este método el 85% de las soldaduras de la cara superior de la placa recorrieron el agujero para llegar a la malla, con lo que las soldaduras que tuve que eliminar por encima fueron mínimas.

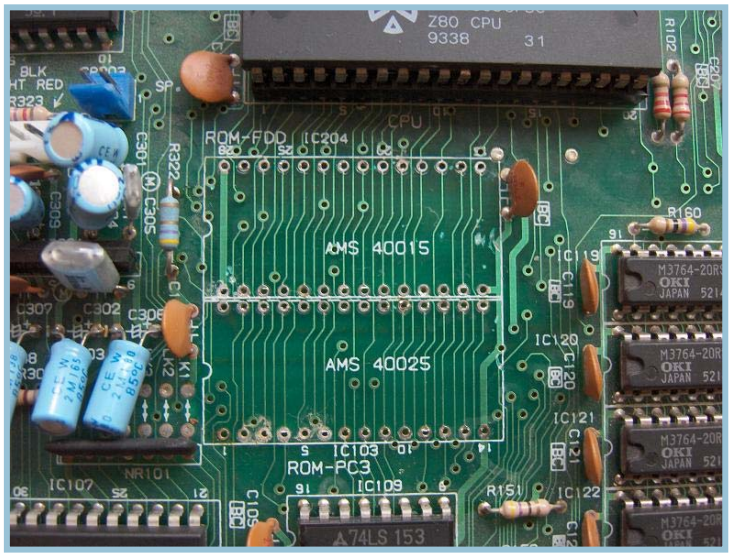
Resumiendo: Elimináis las soldaduras inferiores y luego revisáis las superiores (donde se ven los chips). Si veis que alguna tiene restos de estaño, entonces repetís el proceso por la cara superior de la placa base en cada una de las soldaduras con restos de estaño.

Cuando hayáis terminado con el primer chip empezáis con el siguiente.

En la foto de la derecha podéis comprobar el aspecto de la placa base tras retirar el CHIP del AMSDOS (40015) y los "agujeritos" de las soldaduras que nos sonríen esperando el flamante zócalo que ocupará su lugar.

Os aconsejo que antes de retirar el chip y tras haber eliminado en la medida de lo posible el suficiente estaño de las soldaduras, despeguéis manualmente las patillas de los agujeros. Siempre queda un pequeño resto de soldadura que mantiene unidas las patillas del chip a los agujeritos. Para ello, yo fui empujando las patillas por la parte inferior de la placa base con un destornillador hasta escuchar un pequeño "clack". Eso nos indica que la patilla ya está suelta.

Cuando terminéis de soltar todas las patillas podéis tirar sin miedo del chip y veréis asombrados como sale suavemente de su actual ubicación.



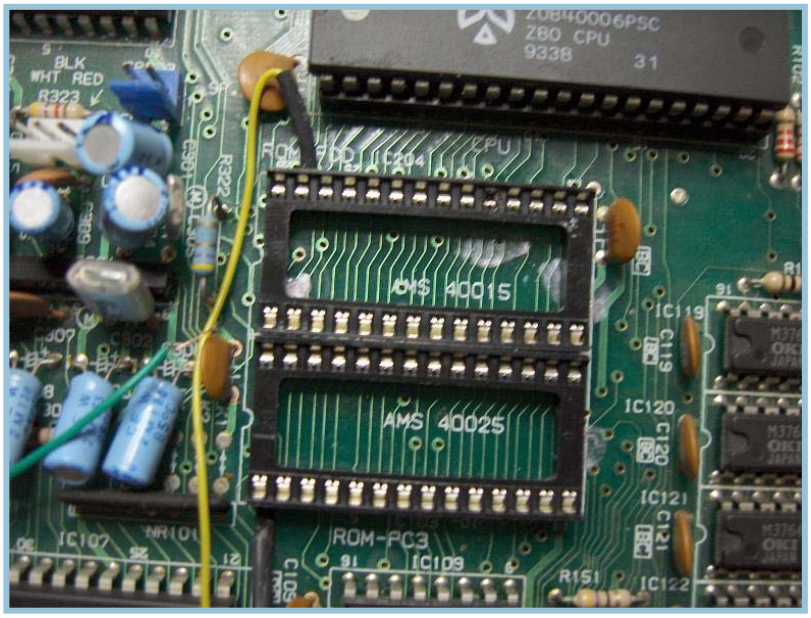
En la imagen de la izquierda podéis observar como queda la placa con ambos chips desoldados.

Si has llegado hasta aquí, felicidades. Acabas de superar la parte más laboriosa del proyecto. Concretamente 1 hora y media como máximo de tiempo invertido. Cuando le cojas práctica, en poco más de media hora puedes haber desoldado media placa base de tu CPC.

Ha llegado el momento de soldar los zócalos donde antes teníamos los chips. Nada más sencillo. Si te fijas en la foto anterior, sobre la placa, en la ubicación de los chips, y bordeando los agujeritos hay 2 recuadros blancos con una "muesca" en la parte izquierda. Esto nos va a indicar la posición del zócalo (y de nuestras EPROMS). Aunque parezca una tontería, realmente es importante mantenerlo así.

Pues bien, podemos continuar. Ahora vamos a insertar los zócalos en los agujeritos PEEEEERO todos menos uno por zócalo. ¿Por qué? Sencillo: como vamos a emplear EPROMS del doble de capacidad, lo que haremos para seleccionar un arranque u otro es dándole positivo o negativo a esas patillas. Realmente la cosa es algo más complicada, pero para el montaje no es necesario saberlo.

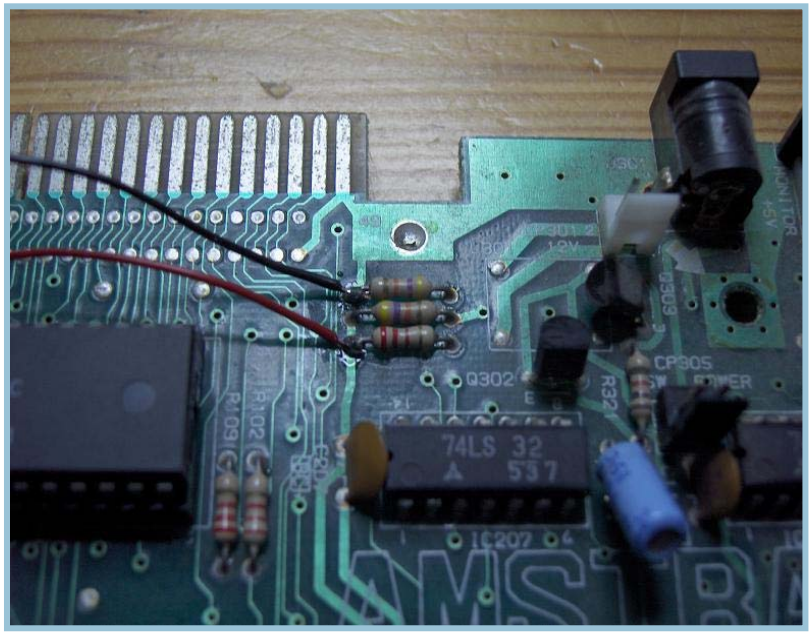
Por si te interesa saberlo, brevemente te explico que se trata manejar el pin de direcciones A14 (en el caso de la EPROM 27C256) y el A15 (en el caso de la EPROM 27C512) para seleccionar la primera mitad o la segunda de las EPROMS.



En la imagen de la izquierda podemos observar ambos zócalos ya soldados en su correspondiente lugar. Fíjate, que en el zócalo superior hay una patilla (la número 27) que no va soldada a la placa y que tiene un cable soldado. Lo mismo ocurre con el zócalo inferior, pero aquí es la patilla número 1 la que dejamos sin soldar y que uniremos con la patilla 27 del primero. Para decirlo fácilmente: La patilla 27 del zócalo superior va unida a la patilla 1 del zócalo inferior.

¡OJO! Hay zócalos con conexión cónica. En este caso es algo más complicado, así que aconsejo emplear zócalos con patilla plana (que se pueden doblar hacia fuera) para poder soldar los cablecillos. Tened cuidado que no hagan contacto con la soldadura de la placa base, ya que aunque no lo hayamos soldado, es posible que al apoyar el zócalo sobre la placa, haga contacto y provoquemos un corto. Yo lo que hice fue emplear laca de uñas (y no soy mariposón) y pinté sobre la soldadora de la placa, para evitar el contacto directo de las patillas dobladas de los zócalos sobre los mismos.

Prosigamos… Ahora vamos a soldar un cablecito rojo y uno negro en un lugar de la placa base. El cablecito rojo nos va a suministrar 5V rectificados (cuidado que no son los mismos 5V del conector de alimentación) y el cable negro nos va a dar 0V (la masa). La masa si podemos cogerla de cualquier punto de la placa, pero yo he preferido cogerla de un lugar cercano al cable de 5V para no dispersar cables de un lado a otro y tenerlo todo lo más ordenado posible.

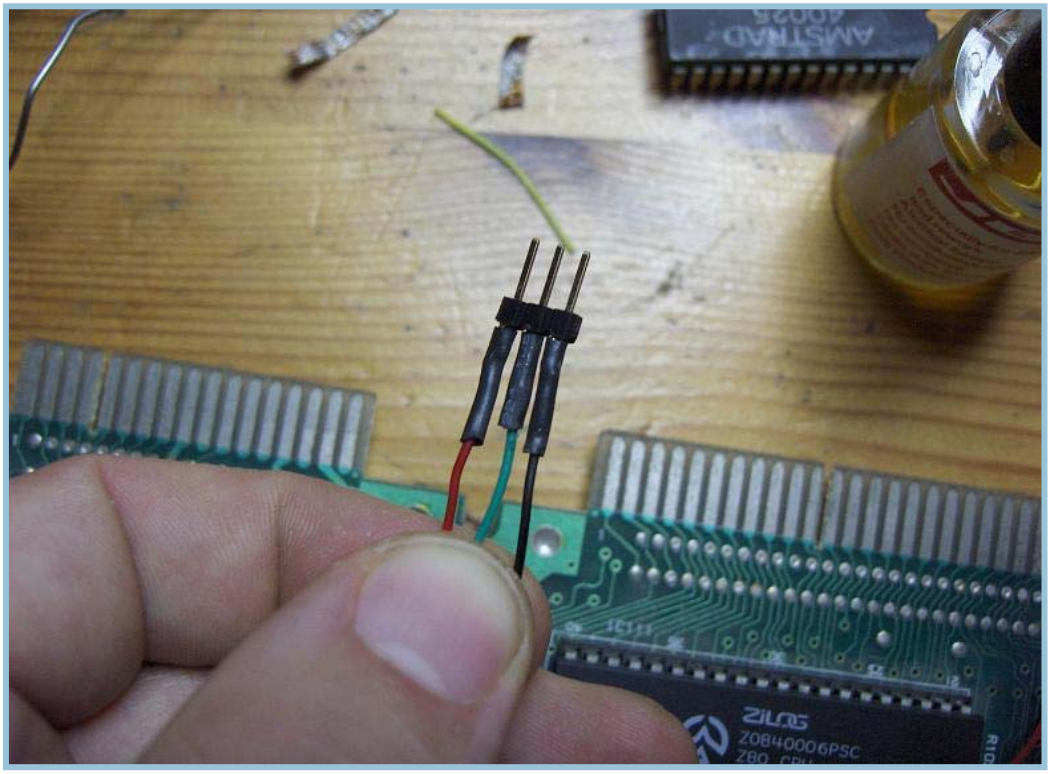


En la foto de la derecha puedes observar el lugar de donde yo he sacado los 5V y la masa (GND en inglés, que queda más internacional). Hay más lugares de donde sacarlos, pero por su proximidad al invento y por su lejanía a otros circuitos integrados importantes he decidido sacar las señales de este lugar. Las resistencias están justo al lado del conector de alimentación (al lado del bus de expansión). El montaje se ha realizado sobre una placa REVISIÓN 1. Es probable que en revisiones más modernas se encuentren en distinto lugar o simplemente NO ESTÉN.

Si se da el caso, tendréis que buscar los 5V que, repito, no son los 5V directos del conector de alimentación. Cualquier pista o componente que vaya a una patilla de alimentación de algún integrado de la placa (procesador, gate array, memorias, o la propia alimentación de las ROMs) tendrán esos 5V que nosotros necesitamos.

Bien, ya estamos terminando la parte "chispas" del asunto. Ahora, tan sólo nos queda soldar el cable del puente de los zócalos y los cables de 5V y masa a los espadines (concretamente 3). Pero con una foto lo veremos todo mucho mejor.

Aquí se ve perfectamente el procedimiento.



Lo único a tener en cuenta es que tenemos que soldar el cable que viene de los zócalos entre los cables de 5V y de masa, como se aprecia en la foto.

Así podremos puentear los zócalos con un jumper a los 5V o a masa, según nos apetezca arrancar con un juego de ROMS o con otro. Pues bien, ya hemos terminado con la parte chispas.

PREPARANDO LAS ROMS

Bien, aquí no voy a extenderme demasiado. Ya he comentado al principio del tutorial que debéis tener los archivos de las ROMs originales y los modificados a punto para grabar en vuestras eproms.

Lo que sí voy a explicar es como crear los archivos para poder grabarlos directamente. Cómo he comentado anteriormente, cada una de las ROMs ocupan 16Kbytes. ¿Cómo puedo generar la ROM de 32Kbytes? Muy fácil:

Imaginemos que tenemos las ROMs de 16Kbytes con los siguientes nombres:

OS.ROM
BASIC.ROM
AMSDOS.ROM

Lo que tenemos que hacer es unir la ROM "OS.ROM" y la ROM "BASIC.ROM" en un mismo fichero y en este mismo orden. Para ello nos iremos a la línea de comandos, nos iremos a la carpeta donde tengamos los archivos de las ROMs y escribiremos:

COPY /B OS.ROM + BASIC.ROM ROML1.ROM

Así hemos creado el archivo "ROML1.ROM" que contiene el sistema operativo y el BASIC originales. El AMSDOS ya lo tenemos en el formato correcto. Ahora tenemos que hacer lo mismo con los archivos ROM personalizados, aquellos donde hayamos incluido nuestros cambios a los originales del sistema, y en el mismo orden: Archivo ROM del OS + archivo ROM del BASIC. Supongamos que hemos creado el archivo "ROML2.ROM". . Ahora tenemos que crear un único archivo que llamaremos "ROMOS.ROM" y que contendrá "ROML1.ROM" y "ROML2.ROM" en este orden, empleando la instrucción "COPY" anterior:

COPY /B ROML1.ROM + ROML2.ROM ROMOS.ROM

Ya tenemos creado nuestro archivo que contiene ROM del sistema operativo y BASIC en formato original y modificado, en este mismo orden y cuyo tamaño es de 64Kbytes (exactamente el tamaño de nuestra EPROM).

Ahora vamos a crear la ROM de disco, a partir de los archivos "AMSDOS.ROM" y "PARADOS.ROM".

COPY /B AMSDOS.ROM + PARADOS.ROM DSK.ROM

Escribimos lo siguiente:
Hemos creado el archivo "DSK.ROM" de 32Kbytes de tamaño, que será el que emplearemos para grabar en la

Grabar ROMOS.ROM en EPROM 27C512
Grabar DSK.ROM en EPROM 27C256

segunda EPROM. Resumiendo:
TERMINANDO Y COMPROBANDO

Pues sí señores, ya estamos terminando. Ahora ya tenemos las ROMS en nuestras EPROMs a buen recaudo (gracias a la tienda de electrónica de turno o al coleguilla con grabador). Finalizada la grabación recomiendo tapar la ventanita de la EPROM con un papelito identificativo (bien con una pegatina o con un poquito de celo) por varios motivos:

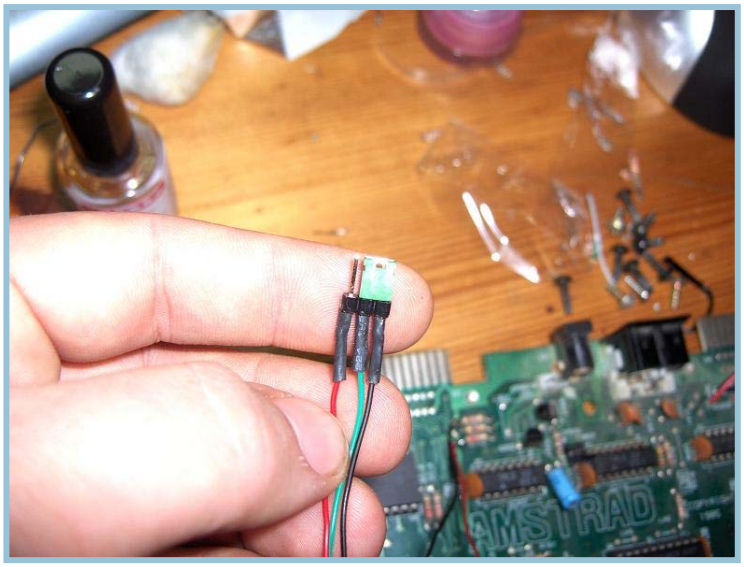
- *Tendremos identificadas las EPROMS a simple vista*
- *Evitaremos que se nos borren impidiendo la entrada de luz solar y/o luz ultravioleta.*

Ya podemos pinchar las memorias en sus correspondientes zócalos (y teniendo en cuenta la posición de la muesca, como en los zócalos) tal cual en la foto que aquí se adjunta:

La que pone "DOS" es la EPROM grabada con el archivo "DSK.ROM" y la que pone "LOWROM" es la EPROM grabada con el archivo "ROMOS.ROM".

Ahora llega el momento de la verdad. Podemos empezar a montar la placa base en su respectivo lugar y conectar el cable del interruptor que viene desde la carcasa. También conectaremos el cable del monitor y el cable de alimentación de 5V (el que viene del monitor). No hace falta que atornillemos la placa (más que nada por si el invento no funciona y tenemos que desmontarlo de nuevo para revisarlo.

Lo siguiente es colocar un bonito "jumper" en los espadines para seleccionar el modo de arranque (normal o chachi). Si el puente une el cablecito rojo (5V) con el cable del centro (el que viene de las EPROMS) estaremos seleccionando las ROMS modificadas (y que están grabadas en la segunda parte de las EPROMs. Si el puente se realiza sobre el cable negro, entonces seleccionamos la primera parte de la memoria de las EPROMS donde hemos grabado las ROMS originales, con lo que el ordenador arrancará como si nada hubiese pasado.



Una foto para explicarlo vale más que mil palabras:

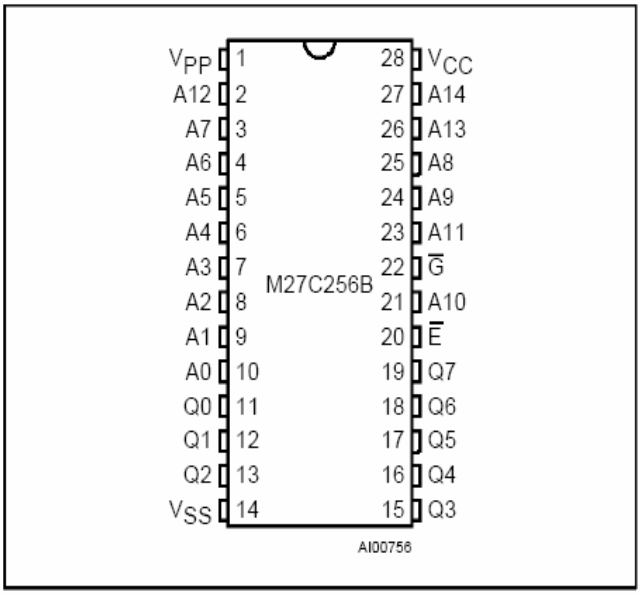
En la foto se puede apreciar que el puente se ha realizado sobre el cable negro (masa) y el cable verde (EPROMS). Si encendemos el invento ahora, nos aparecerá el Amstrad de toda la vida.

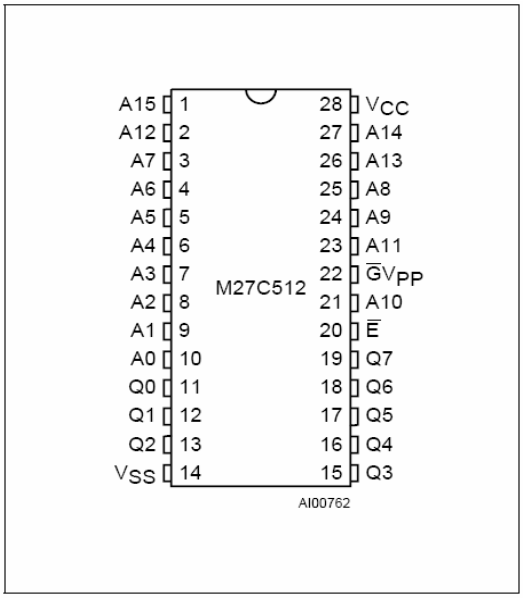
Si cambiamos el puente de sitio y unimos el cable rojo (5V) y el verde y encendemos el ordenador, os sorprenderéis al ver como arranca vuestro maravilloso sistema operativo personalizado con PARADOS, capaz de formatear y manejar disqueteras de PC de 3,5" y hasta 800Kbytes por disquete.

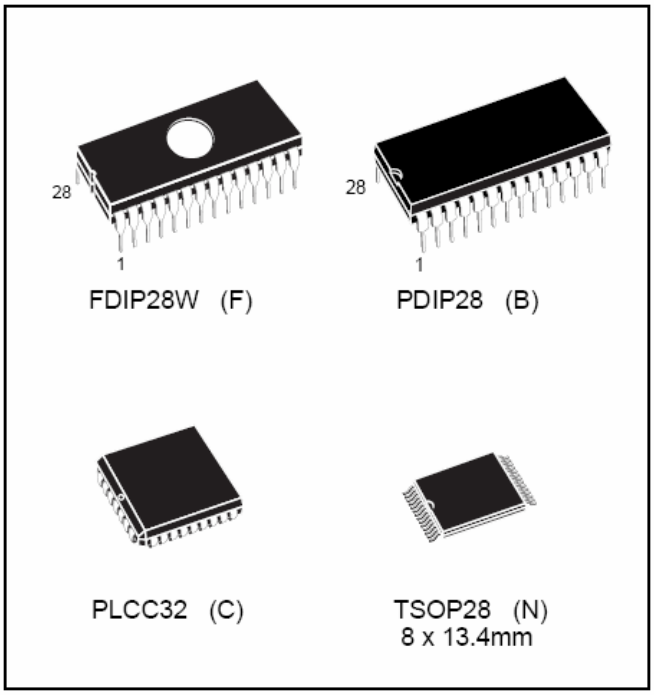
Concretamente, con mis ROMS modificadas, me aparece la siguiente pantalla en el monitor indicándome que todo ha ido como la seda:

¡Felicidades! Acabas de dar a luz tu nuevo Amstrad CPC 6128 DUALBIOS. Menuda satisfacción, ¿verdad? Ya puedes montarlo todo, atornillar y cerrar el invento, que jamás tendrás que volver a abrirlo (no al menos para cambiar las ROMs).

PINOUT (PATILLAJE) DE LAS EPROMS







Recuerda que la parte del puente debes dejarla en el exterior del ordenador para poder conmutar siempre que quieras (con el equipo apagado) el sistema.
También puedes cambiar el puente por un conmutador de 2 posiciones y 3 patillas y agujerear la carcasa y ponerlo ahí, eso ya a gusto del consumidor. Yo he preferido no perforar (por el momento) ningún plástico y dejarlo como en la foto.

Pues nada, espero que este tutorial os haya servido de algo y para animaros a meterle mano a vuestros ordenadores, que aunque tengan más de 20 años aguantan muchísimo más que los actuales, así que perdedle el miedo.

¡Saludos y suerte!

David Donaire Sánchez (DaDMaN)
10 de Abril del 2007

El "Top Ten" de...:

Game Over

Para mí fue todo un impacto visual. Quizás no sea un derroche técnico, sin embargo su aspecto de tonos azules se convertiría en la seña de identidad de los mejores juegos de Dinamic y, de hecho, de la propia compañía. Además el diseño de los gráficos me pareció soberbio . Lo jugaba una y otra vez y nunca me cansaba.

1

Cobra's Arc

Soy consciente de que mis juegos favoritos no son los mejores ni a nivel técnico ni en jugabilidad. Sencillamente son aquellos que me dejaron un buen sabor de boca y que recuerdo con especial afecto. Cobra's Arc es un claro ejemplo. La primera Aventura conversacional que jugué fue Gremmlins y después este juego de Dinamic. Su portada, de Azpiri me cautivó y me metió en la aventura de tal forma que no paré hasta finalizarla.

2

Dustin

Quizás el orden de preferencia de esta lista tampoco diga nada. Es un orden de importancia en cuanto a "vivencias" y recuerdos que cada uno de estos juegos me aportó. Dustin fue uno de ellos. Nunca conseguí terminarlo, ni tan siquiera con el Micromanía delante, siguiendo las pistas paso a paso. Sin embargo no paraba de intentarlo una y otra vez. Sus gráficos y esa "medio 3D", me encantaba.

3

Don Quijote

La publicidad del juego me atrapó irremediablemente. La historia y el personaje hicieron el resto. Creo que fué la primera aventura conversacional que saboreé realmente y, tiene narices, no conseguí terminarla. El problema del tablón en la pared era una pesadilla. Por mucho que soltara siempre tenía exceso de equipaje. Y no hablemos del maldito escalón. La otra gran aventura que me hizo disfrutar como ninguna otra fue...

4

La Aventura Original

Para mí la mejor aventura conversacional realizada en España. Mejor dicho, la mejor versión realizada de la original. Su jugabilidad consiguió que me enganchara definitivamente a este género. Tiene un nivel de dificultad muy inteligente que te invita a llegar un poco más lejos cada día. Y el parser está excelentemente aprovechado.

5

Goody

Es otro de los culpables de hacerme pasar horas delante de mi CPC472. Me encantaba la sensación de poder entrar en el metro de opera, pasar por las barcas del retiro y entrar al Banco de España. Esos toques que Opera hacía para sentirlo más cercano. Al menos a los que vivíamos en Madrid.

6

La Abadía del Crimen

Técnicamente quizá fuera una maravilla, con un buen guión y una excelente trama, sin embargo lo que más me atraía de este juego eran sus gráficos. Me divertía sencillamente deambular de un lado a otro, observando, descubriendo nuevos lugares de la impresionante arquitectura de la abadía. Nunca lo terminé. De hecho me crispaba los nervios muchas situaciones del juego . Me parece endiabladamente difícil.

7

Mad Mix Game

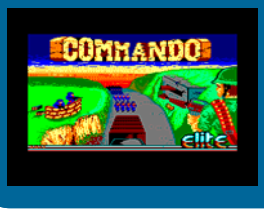
Creo que es, junto a Viaje al Centro de la Tierra, las dos obras maestras de Topo. Es más, creo que son los únicos juegos de Topo que realmente me han gustado. Este come-cocos relanzado es divertido, adictivo, con una gran jugabilidad y te engancha de principio a fin. Creo fue uno de los mejores juegos de la era de los 8 bit, aunque muchos no lo piensen así.

8

Night Shade

Con el debido respeto a KnightLore, por su puesto, la técnica filmation con scroll me dejó anonadado. No concebía que algo así pudiera caber en un CPC y me maravillaba la sensación de libre albedrío por una ciudad medieval. Night Shade me hizo pasar muy buenos ratos aunque… tampoco lo finalice nunca.

9



Commando

Ahora me doy cuenta, llegando al número 10, lo difícil que es hacer una lista tan pequeña teniendo tantos buenos juegos en la cabeza, pero quiero darle una oportunidad a Commando. Era una de las veces que nos encontrábamos lo más cerca posible de tener una recreativa en nuestro ordenador casero y este juego, con su melodía y menú inicial, conseguía hacerlo.

10

Envíanos tu "TOP Ten" a: **amstrad.mail@gmail.com**

El Juego que arrasó en la última edición de MadriSX ahora en tu Amstrad CPC 464—6128.

¡ Amstrad CPC !

PHANTOMAS SAGA INFINITY

Nuevos gráficos. Nuevo sonido. Reprogramado desde cero.
Prepara tu Amstrad CPC para un nuevo clásico

Una producción de

cezgs.computeremuzone.com

También disponible en Spectrum y MSX. Es un producto exclusivo de Computer Emuzone Games Studio. CEZ GS es una división de Computer Emuzone

En números anteriores aprendimos a crear una sencilla base de datos en la que meter información sobre nuestros juegos y a sacar esa información en pantalla. En este capítulo veremos cómo ordenar nuestra base de datos mediante índices y cómo hacer en ella búsquedas y reemplazos.

Y una novedad interesante, gracias a la labor de cpcmaniaco y transformer ha aparecido una versión en castellano de dBASE, así que, a partir de este momento es la versión que utilizaré.

```
CP/M Plus  Amstrad Consumer Electronics plc

v 1.0, 61K TPA, 2 disc drives

A>dbase

ENTRE LA FECHA DE HOY O (RETURN) PARA OMITIRLA
 (DD/MM/AA) :19/02/07

Copyright (C) 1982 RSP Inc.

***  dBASE II     Ver 2.4  1 Abril, 1983

Entre 'HELP', 'HELP dBASE', o un comando

.
```

Ordenando la base de datos

Existen dos posibilidades para ordenar una tabla, la primera es la que a todo el mundo se le puede ocurrir sin pensarlo mucho y es coger los registros y ordenarlos directamente. Esto dBASE lo hace generando una nueva tabla con el orden que nosotros deseemos. La forma de hacerlo es:

```
SORT ON <CAMPO CLAVE> TO <NOMBRE TABLA NUEVA> ASCENDING (O DESCENDING)
```

Por ejemplo, ordenar por el título la tabla de los juegos de CPC:

```
. SORT ON TITULO TO CPC-TIT ASCENDING
SORT COMPLETE
.
```

Pros:
Se tiene la tabla ordenada y hacer un listado de la misma en el orden el que está es muy rápido ya que el acceso a la información es secuencial.
Contras:
Sólo permite el orden por un campo, lo cual puede ser inconveniente si tenemos una tabla con nombres y apellidos y la queremos ordenar.
Si añadimos un registro nuevo se añade al final y casi seguro que no sería esa su posición ordenada.
La ordenación es muy lenta ya que tiene que generar la tabla entera leyendo cada registro y poniéndolo en su sitio.

Se ha generado un nuevo archivo de base de datos llamado CPC-TIT que tiene todos los registros ordenados.

```
00039  Donkey Kong                                  cinta        1
00040  Dragon's Lair II                             cinta        1
00041  Drangonninja                                 cinta        1
00042  Dwarf                                        cinta        1
00043  El laberinto del sultan                      cinta        1
00044  Enduro Racer                                 cinta        1
00045  Fernando Martin                              cinta        1
00046  Flight simulator                             cinta        1
00047  Frank'n'stein                                cinta        1
00048  Fruit Machine                                cinta        1
00049  FutureBike                                   cinta        1
00050  Golden Axe                                   cinta        1
```

La otra forma de ordenar es mediante el uso de los índices

Empezaremos con un poco de chapa para aclarar el uso de índices para conseguir ordenaciones eficientes. Imaginemos una tabla en un disquete, que es al fin y al cabo nuestra base de datos. Queremos coger los registros y ordenarlos por título por ejemplo. Nuestra tabla de ejemplo tiene varios campos: Título, soporte, número de copias,... Si actuáramos contra ella para ordenarla tendríamos que hacer inserciones y borrados de todos los campos de cada registro que se moviera en el disquete, lo cual es realmente lento y se tornaría insufrible si fueran muchos los registros. Ahí entran en juego los índices, a la tabla inicial le añadimos un campo más que es un índice que relaciona todo el registro, que es el número que se muestra a la izquierda de los registros. La forma de utilizar este registro es mediante un fichero de índice en el que se dice cómo se ordenan los índices de la tabla. Por ejemplo, si quisiéramos ordenar por título tendríamos un fichero de registros con el índice y su título y sería esto lo que ordenáramos. A la hora de mostrar la información, dBASE sabría que hay que mostrarla en el orden que indica este fichero y nosotros la veríamos ordenada.
Esto se hace con la orden siguiente:

```
INDEX ON <EXPRESIÓN> TO <NOMBRE ARCHIVO>
```

Donde <expresión> puede ser simplemente un campo o una expresión más elaborada del tipo de las que hemos visto en el capítulo anterior.
Y donde <nombre de archivo> es el archivo del índice ordenado que se va a generar.
El resultado es que la tabla original no se toca, pero al mostrarla usando el índice se ve igualmente ordenada.

```
. INDEX ON TITULO TO TITULO
00100  REGISTROS INDEXADOS
00168  REGISTROS INDEXADOS
.
```

Por ejemplo, INDEX ON TITULO TO TITULO generará un archivo llamado TITULO.NDX que podremos utilizar en nuestra base de datos mediante:

```
USE CPC INDEX TITULO
```
(si la BBDD no está abierta)

```
SET INDEX TO TITULO
```
(si la BBDD ya está abierta)

Se pueden usar hasta 7 índices por BBDD, el uso sería así:
SET INDEX TO <indice1>,<indice2> ,.., <indice7>

Una vez asignado un índice a una Base de datos, los índices serán tenidos en cuenta cada vez que hagamos una inserción, una edición, una lectura o una navegación por los registros. Pero claro, si usamos muchos índice podemos ralentizar mucho el acceso a la BBDD así que moderación.

Buscando en la base de datos.

También existen dos formas de buscar algo en la base de datos. La primera es mediante el comando `FIND <CADENA>` *y se usa sobre bases de datos indexadas. Se posiciona en el primer registro que coincida con la búsqueda realizada, la búsqueda se hace sobre el campo indexado. Una vez posicionado, para ver el registro es necesario hacer un* `DISPLAY`.

Ejemplo:

`FIND WON`
Se posiciona sobre el registro de Wonder Boy

DISPLAY
Muestra el registro.

```
. FIND Won
. DISPLAY TITULO
00162  Wonder Boy
.
```

Nota: Esta búsqueda es la más rápida ya que el archivo índice está ordenado como un árbol binario en el que basta con ir seleccionando la posición izquierda o derecha de cada nodo para llegar a localizar el registro buscado dado que cada hijo cumple una condición excluyente, o es mayor que el padre o es menor.

Y el otro método para buscar es con la orden `LOCATE [<RANGO>] FOR <CADENA>` *también va localizando la cadena en los diferente campos de los registros dentro del rango, no sólo en el índice. La primera vez se posiciona en la primera coincidencia pero se puede seguir buscando con* `CONTINUE`.

```
. LOCATE ALL FOR TITULO="Co"
REGISTRO: 00029
. DISPLAY TITULO FORMATO
00029  Cobat Zone                                    -
. CONTINUE
REGISTRO: 00030
. DISPLAY TITULO FORMATO
00030  Comando                                       -
. CONTINUE
REGISTRO: 00031
. DISPLAY TITULO FORMATO
00031  Confusion                                     Cinta
. CONTINUE
REGISTRO: 00168
. DISPLAY TITULO FORMATO
00168  Columns CPC                                   Cinta
. CONTINUE
FIN DE ARCHIVO ALCANZADO
.
```

Reemplazando datos en la base de datos

Muchas veces es necesario hacer un cambio en un dato o cambiar una palabra por otra. Para estos casos dBASE tiene una pequeña solución:
Es decir, reemplaza en un rango, el campo por la expresión.

```
REPLACE [<RANGO>] <CAMPO> WITH <EXP> [,<CAMPO> WITH <EXP>] [FOR <EXP>]
[NOUPDATE] [WHILE <EXP>]
```

Se puede ampliar y actualizar más campos con otras expresiones y se puede limitar con otros criterios.

Un ejemplo sencillo sería:

```
REPLACE ALL SOPORTE WITH "CINTA" FOR SOPORTE="CINTA"
```

Que cambiaría todos los registros en los que el campo SOPORTE es cinta y pondría Cinta.
Si la base de datos es indexada y se actualiza el índice, se actualiza el indexado pero si no queremos que ocurra se añadiría la opción NOUPDATE.

Si solo hubieramos querido actualizar los registros en los cuales nos indica que tenemos dos copias habríamos podido hacer:

```
REPLACE ALL SOPORTE WITH "CINTA" FOR SOPORTE="CINTA" WHILE COPIAS=2
```

El siguiente ejemplo cambia cinta por Cinta en los títulos que empiezan por W:

```
. LIST TITULO SOPORTE FOR TITULO="W" .AND. SOPORTE="cinta"
00159  Willow Pattern                                    cinta
00160  Winter Games                                      cinta
00161  Winter Games                                      cinta
00162  Wonder Boy                                        cinta
. REPLACE ALL SOPORTE WITH "Cinta" FOR TITULO="W" .AND. SOPORTE="cinta"
00004 REEMPLAZO(S)
. LIST TITULO SOPORTE FOR TITULO="W"
00158  WEC Lemans                                        disco
00159  Willow Pattern                                    Cinta
00160  Winter Games                                      Cinta
00161  Winter Games                                      Cinta
00162  Wonder Boy                                        Cinta
00163  Wonder Boy                                        disco
.
```

Y para terminar, el comando CHANGE que se puede considerar como un REPLACE interactivo ya que va recorriendo los registros que se van a cambiar y permite hacer modificaciones. El formato es:

```
CHANGE [<RANGO>] FIELD <LISTA> [FOR <EXP>]
```

Como ejemplo:

```
CHANGE ALL FIELD SOPORTE FOR TITULO="COBAT"
```

dBASE va mostrando los registros que cumplen la condición y en cada uno solicita la modificación que se quiere hacer.

```
. CHANGE ALL FIELD TITULO FOR TITULO="Cobat"

REGISTRO: 00029


TITULO: Cobat Zone
CHANGE? Cobat
TO       Combat

TITULO: Combat Zone
CHANGE?
.
```

Cuando pregunta CHANGE? Se le indica las letras a sustituir, y en TO lo que se quiere poner. Podríamos habar cambiado simplemente b por mb, por ejemplo. El resultado habría sido el mismo.

Esto es todo por ahora, en el próximo número haremos algún informe con los datos que tenemos. Y por supuesto, si hay algo que queráis aclarar o ampliar lo podéis hacer en el buzón de la revista.

No termina aquí y para continuar necesita de tu colaboración.
Artículos, Revisiones, Comentarios, Entrevistas, Imágenes, Ideas, Ayudas.
¿ Quieres colaborar ?

amstrad.mail@gmail.com
Foro www.amstrad.es/forum